



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA

DESARROLLO DE UN
SISTEMA PARA EL
REGISTRO Y CONTROL DE
ACTIVIDADES EN SALAS
DE CINE

Autor: Hector Molla Ruiz

Tutor: Ángel García Crespo

Leganés, Septiembre de 2013

Título: DESARROLLO DE UN SISTEMA PARA EL REGISTRO Y CONTROL DE ACTIVIDADES EN SALAS DE CINE

Autor: Hector Molla Ruiz

Director: Ángel García Crespo

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Han sido muchos años llenos de buenos y no tan buenos momentos, pero aquí estoy, a punto de alcanzar la meta. Son muchas las experiencias, conocimientos y personas que me llevo de esta etapa de mi vida que está a punto de finalizar.

Agradezco a mi tutor Ángel García por la confianza depositada en mí y por brindarme la oportunidad de realizar este proyecto. Sin duda ha sido el final perfecto para mi etapa universitaria.

A Israel, Jose Luis y el resto del equipo de Whatscine, por hacerme sentir parte del grupo desde el principio.

A las dos personas que han hecho posible que haya llegado donde estoy, mis padres. Gracias por vuestro apoyo incondicional, por confiar en mí incluso cuando menos lo merecía. Soy quien soy gracias a vosotros.

A mis compañeros y amigos, por la cantidad de experiencias vividas. A Lucia, por ser mi faro cuando estaba perdido, a Pablo, por acompañarme a lo largo de este viaje y a Ana, por enseñarme un nuevo mundo y proporcionarme la motivación que necesitaba.

Gracias.

Resumen

Whatscine es un sistema diseñado para proporcionar accesibilidad, entretenimiento y nuevas vías de publicidad dentro de las salas de cine. El sistema se compone de varios elementos, entre los que destaca el que podría considerarse como su núcleo, la aplicación para dispositivos móviles iOS y Android. Es esta la que proporciona a los usuarios el acceso a todos los servicios.

El presente proyecto consiste en el desarrollo, para la versión de Android, de un sistema que se encarga de realizar el control y el registro de la actividad de los usuarios en las salas de cine, así como de la codificación, entrega y consulta de los premios otorgados. Alcanzar estas funcionalidades implica el desarrollo de un módulo para la obtención de la información en los dispositivos móviles y la implementación de un servidor propio para la gestión y almacenamiento de la misma.

Palabras clave: Android, Java, cliente, servidor, base de datos, información, actividad, usuario, premio, desarrollo.

Abstract

Whatscine is a system designed to provide accessibility, entertainment and new ways of advertising in cinemas. The system consists on several elements, being the most important among them the application for iOS and Android mobile devices. This application gives user access to all the services.

The project involves the development of a system responsible for the control and registration of user's activity in movie theaters, as well as the coding, assignment and query about the prizes. It requires the development of a module to obtain the necessary information from mobile devices, and also the implementation of a dedicated server for the storage and management of relevant data.

Keywords: Android, Java, client, server, database, information, activities, user, award, development.

Índice general

1. INTRODUCCIÓN Y OBJETIVOS.....	1
1.1 Introducción	1
<i>Servidores.....</i>	<i>1</i>
<i>Dispositivos móviles.....</i>	<i>2</i>
<i>Sistemas operativos.....</i>	<i>4</i>
<i>Aplicaciones móviles.....</i>	<i>4</i>
<i>Whatscine</i>	<i>5</i>
1.2 Objetivos	8
1.3 Medios empleados.....	8
<i>Software</i>	<i>8</i>
<i>Hardware</i>	<i>9</i>
1.4 Estructura de la memoria	10
2. TECNOLOGÍAS Y PLATAFORMAS	11
2.1 Dispositivos móviles	11
<i>Android.....</i>	<i>11</i>
2.2 Servidor Whatscine	16
<i>Java</i>	<i>16</i>
<i>Web service</i>	<i>16</i>
<i>JSP</i>	<i>17</i>
<i>HTML.....</i>	<i>17</i>
<i>CSS.....</i>	<i>18</i>
<i>MySQL</i>	<i>18</i>
<i>Apache Tomcat.....</i>	<i>18</i>
2.3 Servidor sala.....	19
<i>C</i>	<i>19</i>
2.4 Entornos de desarrollo.....	19
<i>NetBeans IDE.....</i>	<i>19</i>
<i>Eclipse.....</i>	<i>20</i>
<i>Xcode.....</i>	<i>20</i>
3. FUNCIONALIDAD Y TAREAS DEL SISTEMA.....	21
3.1 Tarea 1: Registro actividad usuarios	21

ÍNDICE general

3.2 Tarea 2: Usuario premiado	24
3.3 Tarea 3: Canjear código premiado	27
3.4 Tarea 4: Gestión salas de cine	29
4. ANÁLISIS DEL SISTEMA.....	31
4.1 Arquitectura general	31
4.2 Servidor Whatscine	33
<i>Servicio Web</i>	33
<i>Página Web</i>	35
<i>Base de datos</i>	39
<i>Núcleo</i>	44
4.3 Servidor sala	49
4.4 Cliente Android	51
<i>Clases</i>	52
<i>Interfaz</i>	55
<i>Permisos</i>	55
<i>SDK</i>	55
4.5 Pruebas	57
<i>Algoritmo de generación de códigos</i>	57
<i>Página Web</i>	57
<i>Aplicación Android + Servidor Whatscine</i>	61
5. PRESUPUESTO	66
5.1 Fases de desarrollo	66
<i>Fase 1: Servidor Whatscine</i>	66
<i>Fase 2: Cliente Android</i>	67
<i>Fase 3: Servidor Sala</i>	67
<i>Fase 4: Sistema Completo</i>	68
5.2 Desglose de costes.....	70
6. CONCLUSIONES Y LÍNEAS FUTURAS	72
6.1 Conclusiones	72
6.2 Líneas futuras	73
7. GLOSARIO	75
8. REFERENCIAS	76

Índice de figuras

Figura 1. Logo Whatscine[8].	5
Figura 2. Ilustración de distintos métodos de accesibilidad[8].	6
Figura 3. Ilustración uso aplicación[8].	6
Figura 4. Diagrama sistema Whatscine[8].	7
Figura 5. Arquitectura interna Android[10].	12
Figura 6. Ciclo de vida aplicación Android[10].	14
Figura 7. Escenario tarea 1.	22
Figura 8. Diagrama de flujo tarea 1.	23
Figura 9. Escenario tarea 2.	25
Figura 10. Diagrama de flujo tarea 2.	26
Figura 11. Escenario tarea 3.	27
Figura 12. Diagrama de flujo tarea 3.	28
Figura 13. Escenario tarea 4.	29
Figura 14. Diagrama tarea 4.	30
Figura 15. Arquitectura general del sistema completo.	32
Figura 16. Bloques servidor Whatscine.	33
Figura 17. Arquitectura general página web.	36
Figura 18. Vista página inicio de sesión.	36
Figura 19. Vista página registro de usuario.	37
Figura 20. Vista página información salas de cine.	38
Figura 21. Vista página modificar.	38
Figura 22. Vista página gestión premios.	39
Figura 23. Modelo entidad relación base de datos.	40
Figura 24. Código conexión base de datos.	43
Figura 25. Ejemplo código consulta base de datos.	44
Figura 26. Propiedades servicio Gmail.	48
Figura 27. Diagrama del algoritmo de generación de códigos.	50
Figura 28. Estructura de clases del módulo Android.	52
Figura 29. SDK Android.	56

ÍNDICE DE FIGURAS

Figura 30. Vista aplicación de prueba.....	62
Figura 31. Ejemplo de correo recibido premiado.....	64

Índice de tablas

Tabla 1. Cuota sistemas operativos en España[6].	4
Tabla 2. Programas utilizados en el proyecto.	9
Tabla 3. Hardware utilizado en el proyecto.	9
Tabla 4. Datos necesarios para el registro.	24
Tabla 5. Parámetros método conexión3G.	34
Tabla 6. Parámetros método conexiónWIFI.	34
Tabla 7. Parámetros método correo.	35
Tabla 8. Atributos tabla cine.	41
Tabla 9. Atributos tabla sala.	41
Tabla 10. Atributos tabla usuario.	41
Tabla 11. Atributos tabla actividad.	41
Tabla 12. Atributos tabla premio.	42
Tabla 13. Clase ConectarBD.	46
Tabla 14. Clase Mail.	48
Tabla 15. Parámetros de configuración librería ksoap2.	54
Tabla 16. Permisos necesarios para el módulo Android.	55
Tabla 17. P-01.	57
Tabla 18. P-02.	58
Tabla 19. P-03.	58
Tabla 20. P-04.	58
Tabla 21. P-05.	58
Tabla 22. P-06.	58
Tabla 23. P-07.	58
Tabla 24. P-08.	58
Tabla 25. P-09.	59
Tabla 26. P-10.	59
Tabla 27. P-11.	59
Tabla 28. P-12.	59
Tabla 29. P-13.	59

ÍNDICE DE TABLAS

Tabla 30. P-14.	59
Tabla 31. P-15.	60
Tabla 32. P-16.	60
Tabla 33. P-17.	60
Tabla 34. P-18.	60
Tabla 35. P-19.	60
Tabla 36. P-20.	60
Tabla 37. P-21.	61
Tabla 38. P-22.	61
Tabla 39. P-23.	61
Tabla 40. P-24.	61
Tabla 41. P-25.	61
Tabla 42. P-26.	63
Tabla 43. P-27.	63
Tabla 44. P-28.	63
Tabla 45. P-29.	63
Tabla 46. P-30.	63
Tabla 47. P-31.	63
Tabla 48. P-32.	63
Tabla 49. P-33.	64

Capítulo 1

Introducción y objetivos

1.1 Introducción

Este primer capítulo servirá para situar el proyecto dentro de su contexto. El presente proyecto fin de carrera consiste en un conjunto de bloques que se integrarán dentro del sistema Whatscine, añadiéndole nuevas funciones. De esta forma voy a comenzar dando una introducción general a los dispositivos móviles y servidores, sobre los que funciona el sistema, y haré una breve descripción de Whatscine. Finalmente se dará paso a las tecnologías, plataformas y medios a los que he recurrido como desarrollador.

Servidores

En informática, un servidor es un tipo de software que realiza ciertas tareas en nombre de los usuarios. El término servidor ahora también se utiliza para referirse al ordenador físico (hardware) en el cual funciona ese software, una máquina cuyo propósito es proveer datos de modo que otras máquinas puedan utilizar esos datos. Este uso dual puede llevar a confusión. Por ejemplo, en el caso de un servidor web, este término podría referirse a la máquina que almacena y maneja los sitios web, y en este sentido es utilizada por las compañías que ofrecen hosting u hospedaje. Alternativamente, el servidor web podría referirse al software, como el servidor de http de Apache, que funciona en la máquina y maneja la entrega de los componentes de las páginas web como respuesta a peticiones de los navegadores de los clientes.

CAPÍTULO 1: INTRODUCCIÓN Y OBJETIVOS

Los archivos para cada sitio de Internet se almacenan y se ejecutan en el servidor. Hay muchos servidores en Internet y muchos tipos de servidores, pero comparten la función común de proporcionar el acceso a los archivos y servicios.

Un servidor facilita información a los ordenadores que se conecten a él. Cuando los usuarios se conectan a un servidor pueden acceder a programas, archivos y otra información del servidor.

En la web, un servidor web es un ordenador que usa el protocolo http para enviar páginas web al ordenador de un usuario cuando el usuario las solicita.

Los servidores web, servidores de correo y servidores de bases de datos son a lo que tiene acceso la mayoría de la gente al usar Internet[1].

A continuación se van a citar los principales tipos de servidor que existen[2]:

- Servidor de archivos: es aquel que almacena y sirve ficheros a equipos de una red.
- Servidor de Directorio Activo/Dominio: es el que mantiene la información sobre los usuarios, equipos y grupos de una red.
- Servidor de Impresión: se encarga de servir impresoras a los equipos cliente y poner en la cola los trabajos de impresión que estos generan.
- Servidor de Correo: se encarga de gestionar el flujo de correo electrónico de los usuarios, envía, recibe y almacena los correos de una organización.
- Servidor de Fax: gestiona el envío, recepción y almacenamiento de los faxes.
- Servidor Proxy: su principal función es guardar en memoria caché las páginas web a las que acceden los usuarios de la red durante un cierto tiempo, de esta forma las siguientes veces que estos acceden al mismo contenido, la respuesta es más rápida.
- Servidor Web: Almacena contenido web y lo pone al servicio de aquellos usuarios que lo solicitan.
- Servidor de Base de Datos: es aquel que provee servicios de base de datos a otros programas o equipos cliente.
- Servidor DNS: permite establecer la relación entre los nombres de dominio y las direcciones IP de los equipos de una red.
- Servidor DHCP: este dispone de un rango de direcciones con el cual, asigna automáticamente los parámetros de configuración de red IP a las máquinas cliente cuando estas realizan una solicitud.
- Servidor FTP: su función es permitir el intercambio de ficheros entre equipos, normalmente su aplicación va muy ligada a los servidores Web.

Dispositivos móviles

Los dispositivos móviles son aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras tareas complementarias[3].

Entre las diferentes prestaciones y características que presentan estos nuevos dispositivos, se encuentran:

- Una amplia mejora del almacenamiento de datos.
- Conexión a Internet mediante una tarifa contratada o haciendo uso de redes WIFI.

1.1 INTRODUCCIÓN

- Acelerómetro.
- Pantalla táctil.
- Teclado QWERTY.
- Navegación GPS.
- Multitarea.
- Gran cantidad de aplicaciones de usuario.

Todas estas prestaciones y características de los dispositivos móviles estarían desaprovechadas sin software que las saque partido. Por ello estos dispositivos llevan un sistema operativo que les permite realizar todas estas tareas de una forma rápida y sencilla.

A pesar de estas importantes mejoras con respecto a sus predecesores móviles, el reducido tamaño conlleva inexorablemente algunas limitaciones de hardware que los diferencian de los ordenadores convencionales. Estas limitaciones se ven reflejadas principalmente en pantallas mas pequeñas, menor capacidad de procesamiento, memoria RAM y memoria persistente, y la necesidad de adaptar el consumo de energía a la capacidad de una pequeña batería.

Para el objetivo que nos atañe podemos centrarnos en dos tipos concretos de dispositivos móviles.

Smartphone

Un Smartphone (cuya traducción sería “teléfono inteligente”) es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos, realizar actividades, semejantes a una mini computadora, y conectividad que un teléfono móvil convencional. El término “inteligente” hace referencia a la capacidad de usarse como un ordenador de bolsillo, llegando incluso a remplazar a un ordenador personal en algunos casos[4].

Tablet

Una Tablet es una computadora portátil de mayor tamaño que un Smartphone, integrada en una pantalla táctil (sencilla o multitáctil) con la que se interactúa primariamente con los dedos o una pluma stylus (pasiva o activa), sin necesidad de teclado físico ni ratón[5].

La Tablet funciona como una computadora, solo que más ligera en peso y más orientada al multimedia, lectura de contenidos y a la navegación web que a usos profesionales.

Estos dispositivos móviles se componen de dos partes claramente diferenciadas, su parte Hardware y su parte Software. Es esta última la que nos interesa explicar en detalle, ya que será en la que se base el desarrollo de este proyecto.

En el siguiente apartado, realizaremos una breve descripción de los sistemas operativos para dispositivos móviles más importantes que se encuentran en el mercado a día de hoy.

Sistemas operativos

Los sistemas operativos móviles se encargan de controlar los dispositivos móviles, al igual que ocurre en los ordenadores comunes, sin embargo, estos son mucho más simples y están más orientados a la conectividad inalámbrica, los formatos multimedia para móviles y las diferentes maneras de introducir información en ellos.

Los sistemas operativos móviles más frecuentes utilizados por los teléfonos inteligentes son Android (Google Inc., CA, Estados Unidos), iOS (Apple Inc, CA, Estados Unidos), Symbian (Nokia, Finlandia), BlackBerry OS (BlackBerry, ON, Canadá), y Windows Phone (Microsoft Corporation, WA, Estados Unidos).

Según Kantar Worldpanel ComTech, que ha hecho un estudio sobre las cuotas de mercado de los sistemas operativos en los países más importantes del mundo[6], en España el 89,9% de los Smartphone vendidos en los últimos meses utilizan Android como sistema operativo, siendo así el país con mayor proporción de Smartphone Android en el mercado de los últimos meses. Como dato de referencia tenemos que en el conjunto de los cinco principales países de Europa, el número de Smartphone vendidos supone un 70,4% del total. En la Tabla 1, pueden verse los resultados del estudio referentes a la evolución en la cuota de los sistemas operativos sobre nuevos Smartphone en España.

%Cuota Sistemas Operativos sobre nuevos Smartphones	May - Jul 2012	May - Jul 2013
Android	82,5%	89,9%
iOS	2,9%	6,1%
Symbian	2,1%	0,7%
BlackBerry	9,4%	0,7%
Windows	1,7%	1,8%
Otros	1,4%	0,8%
Total Smartphones	100%	100%

Fuente: Worldpanel ComTech

Tabla 1. Cuota sistemas operativos en España[6].

Aplicaciones móviles

Una aplicación móvil o App es una aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución, operadas por las compañías propietarias de los sistemas operativos móviles.

El desarrollo de aplicaciones para dispositivos móviles requiere tener en cuenta las limitaciones de estos dispositivos. Estos funcionan con batería y tienen procesadores menos poderosos que los ordenadores personales, además, tienen también que considerarse una gran variedad de tamaños de pantalla, datos específicos de software y configuraciones. El desarrollo de aplicaciones móviles requiere el uso de entornos de desarrollo integrados. Las aplicaciones móviles suelen ser probadas primero usando emuladores y más tarde se ponen en el mercado en periodo de prueba[7].

Whatscine

Whatscine es una forma de entretenimiento social totalmente nueva en las salas de cine, una innovadora tecnología que convierte los cines en accesibles por completo; que permite a los espectadores de una sala de cine interactuar con sus teléfonos móviles, Tablet y otros dispositivos, con otros espectadores en tiempo real y posibilita también la emisión de publicidad interactiva por parte de la sala[8]. La Figura 1 muestra uno de los logotipos de la aplicación.



Figura 1. Logo Whatscine[8].

Accesibilidad

Los cines son espacios de ocio cultural de gran consumo, donde asisten con asiduidad familias con niños, jóvenes y personas mayores, en todas las ciudades y pueblos. Pero las personas que sufren alguna discapacidad, no pueden disfrutar de este ocio por la falta de accesibilidad.

Whatscine convierte la sala de cine en accesible, permitiendo a la personas con discapacidad visual escuchar la audio descripción de la película sin interferir en el audio del resto de los espectadores, y a la personas con discapacidad auditiva, ver el lenguaje de signos a través de unas gafas especiales, o seguir el subtítulo en su Smartphone.

Al ser una aplicación multi-idioma, permitirá a la población extranjera disfrutar de esta experiencia gracias a los subtítulos en su idioma nativo. Gracias a esta posibilidad se fomenta la inclusión, el ocio compartido y el acceso para todos a la cultura en igualdad.

En la Figura 2, se muestran gráficamente los distintos métodos de accesibilidad, nombrados anteriormente, que ofrece el sistema de Whatscine.



Figura 2. Ilustración de distintos métodos de accesibilidad[8].

Interactividad

Whatscine permite interactuar a grandes audiencias con la gran pantalla del cine, a través de una aplicación gratuita. Simplemente abrir la aplicación y listo para empezar.

Se pueden crear juegos fácilmente personalizables para cada sala o sesión, adaptando cada uno a la audiencia que asistirá a la proyección. Juegos, desafíos y otras experiencias interactivas con y contra la familia, amigos o la audiencia permitirán ganar premios y recibir información de interés. En la Figura 3 se muestra una ilustración de cómo sería el uso de uno de los juegos en las salas de cine.



Figura 3. Ilustración uso aplicación[8].

1.1 INTRODUCCIÓN

Publicidad

La aplicación también permite la emisión de publicidad interactiva por parte de la sala, facilitando así la promoción de futuras películas y ofertas especiales relacionadas con la sala y sus patrocinadores. Se crean experiencias bidireccionales con las marcas. El espectador ya no es un receptor pasivo de imágenes, texto, vídeo, etc., sino que ahora pasa a un papel mas activo, interaccionando con la publicidad de las marcas, a través de videojuegos, disfrutando y sumergiéndose en una relación mas intensa, directa y personal con el anunciante.

Permite personalizar fácilmente cualquier animación que la marca requiera integrar, convirtiendo la sala de cine en un nuevo canal de comunicación directa con futuros consumidores de sus productos.

Whatscine se integra perfectamente con su TMS, permitiendo programar los juegos interactivos en pre-show, integrado con su planificación de la sala. El sistema de sincronización para la accesibilidad de las películas, solo necesita del canal de audiodescripción de su DCP.

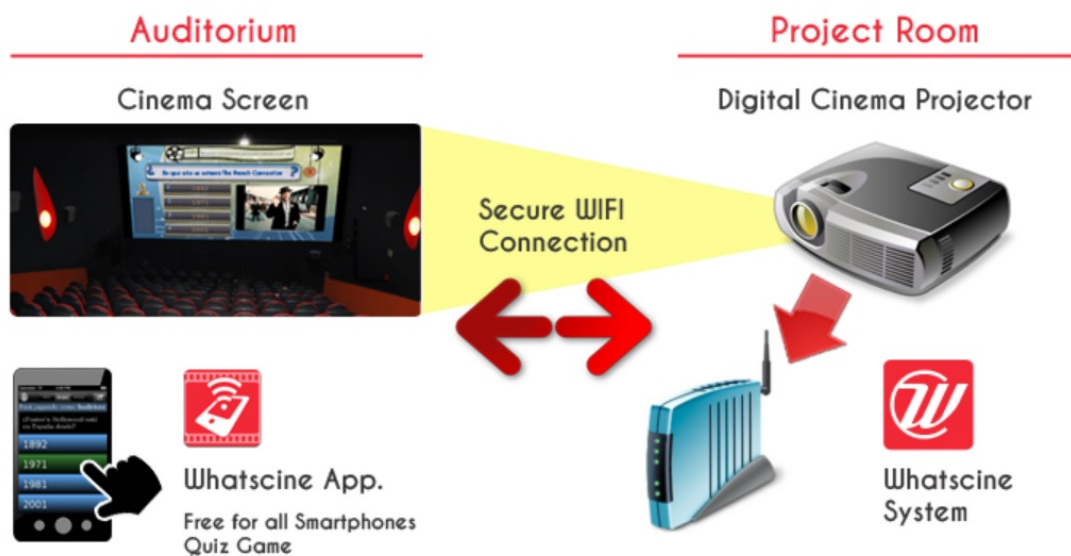


Figura 4. Diagrama sistema Whatscine[8].

La arquitectura del sistema Whatscine, dentro de las salas de cine, puede apreciarse en la Figura 4. Los dispositivos móviles, con la aplicación instalada, se conectan por WIFI al servidor de cada sala mediante routers instalados en las salas de proyección.

La instalación es sumamente sencilla, conectar y listo. Las aplicaciones gratuitas, disponibles tanto para iOS como para Android, permitirán a los espectadores divertirse durante la espera del comienzo de la película, generando ingresos para la sala gracias a la publicidad.

La tecnología de la Universidad Carlos III de Madrid, hace posible que Whatscine sea una empresa enfocada en las técnicas mas avanzadas, para crear ambientes interactivos a través de las últimas y más novedosas tecnologías de inteligencia artificial y tratamiento de entornos reales.

1.2 Objetivos

El objetivo de este proyecto es el desarrollo de un sistema encargado de gestionar y controlar la actividad en las salas de cine, utilizando para ello los conocimientos y aptitudes adquiridos durante toda la carrera.

El proyecto se engloba dentro del sistema Whatscine. Este sistema se encuentra en plena expansión, siendo necesario el desarrollo de componentes que le proporcionen nuevas funcionalidades. Es de aquí donde surge la idea para el proyecto.

La funcionalidad básica del sistema a implementar es:

- Registrar las visitas de usuarios de las salas de cine adheridas al sistema Whatscine.
- Dar soporte para la gestión de las salas adheridas al sistema.
- Controlar los premios proporcionados a través de Whatscine.

Los distintos componentes que son necesarios para el correcto funcionamiento del sistema se encontrarán localizados en tres elementos, el dispositivo móvil, el servidor de las salas y el servidor propio.

Para el dispositivo móvil, la implementación se basa en un módulo para recoger información y enviarla a los servidores. En cuanto al servidor de las salas, se elabora un algoritmo secreto, que genera los códigos para los premios. Finalmente, para lograr los objetivos requeridos, es necesario implementar por completo el servidor propio de Whatscine, dentro del cual se elaboran varios bloques entre los que se encuentran: un servicio web para comunicación con los dispositivos móviles, una página web para manejo por parte de las salas de cine y una base de datos para almacenar la información.

1.3 Medios empleados

Para el desarrollo de este proyecto ha sido necesario el uso de un gran abanico de tecnologías diferentes. El sistema esta compuesto de varias partes, cada una de las cuales esta desarrollada en un lenguaje de programación diferente. Además ha sido necesario el uso de otros lenguajes como complemento para implementar algunas funciones. Por este motivo durante el desarrollo se han utilizado diversos programas y entornos, eligiendo para cada componente el que mejor se ajustara a sus características.

En cuanto a la parte Hardware, son varios los dispositivos que han sido necesarios para el desarrollo del proyecto.

Software

A continuación se especifican cuales han sido las tecnologías, lenguajes y herramientas software de desarrollo que se han utilizado para la implementación del proyecto.

1.3 MEDIOS empleados

Tecnologías, lenguajes y plataformas

Durante el desarrollo de este proyecto son varias las tecnologías, plataformas o lenguajes de programación utilizados. A continuación podemos ver una lista de todos ellos, siendo en el capítulo 2 donde se expliquen más en detalle:

- Android
- Java
- C
- HTML
- CSS
- JSP
- Web Service
- Apache Tomcat

Herramientas

En la Tabla 2 podemos ver las herramientas y programas utilizados en el proyecto, junto con información relevante sobre ellos.

Programa	Desarrollador	Versión	Descripción
NetBeans IDE	Sun Microsystems/Oracle Corporation	7.2	Entorno desarrollo
Eclipse	Eclipse Foundation	4.2	Entorno desarrollo
Xcode	Apple Inc.	4.6.3	Entorno desarrollo
Project	Microsoft	2010	Gestor de proyectos
Erwin	CA technologies	Standard	Diseño bases de datos

Tabla 2. Programas utilizados en el proyecto.

Hardware

En la Tabla 3, se encuentran reflejadas las distintas herramientas hardware que se han utilizado en el desarrollo del proyecto.

Dispositivo	Fabricante	Características	Usos
Macbook Pro	Apple	-Pantalla 15" -Intel Core i7 2.3GHz -8 GB RAM -OS X 10.8	-Desarrollo proyecto -Funcionamiento como servidor Whatscine -Funcionamiento como servidor MySQL.
Samsung Galaxy S3	Samsung	-Pantalla 4,8" -Procesador 4 núcleos 1,4GHz -1 GB RAM -Android 4.1.2	-Pruebas aplicación Android
Samsung Galaxy Ace	Samsung	-Pantalla 3,5" -Procesador 800 MHz -256 MB RAM -Android 2.2	-Pruebas aplicación Android.

Tabla 3. Hardware utilizado en el proyecto.

1.4 Estructura de la memoria

El presente apartado tiene como objetivo explicar, de forma breve y concisa, el contenido de cada uno de los capítulos del documento, facilitando de esta forma su lectura:

Capítulo 1: Presenta el contexto general en el que se ha desarrollado el proyecto. Introduce las tecnologías y medios utilizados, y explica cual ha sido la motivación y los objetivos para su realización.

Capítulo 2: Hace una descripción de las distintas tecnologías y plataformas que se han utilizado para el desarrollo del proyecto.

Capítulo 3: Muestra detalladamente cuales son las funciones y tareas que realiza el sistema. Además, por medio de diagramas, nos enseña las diferentes situaciones de uso real en las que se va a encontrar el sistema.

Capítulo 4: Explica desde un punto de vista técnico el sistema desarrollado en el proyecto. Analiza tanto la arquitectura general del sistema como las distintas partes que lo componen, detallando como se ha realizado su implementación.

Capítulo 5: Incluye un resumen de las diferentes fases por las que ha pasado el proyecto durante su desarrollo y realiza un desglose de los distintos costes que lleva asociada su implementación.

Capítulo 6: Presenta las conclusiones obtenidas al finalizar el proyecto y una serie de posibles líneas futuras, por las que se podría continuar desarrollando el proyecto.

Capítulo 7: Incluye, para una mejor comprensión, la explicación de algunos términos utilizados en la documentación.

Capítulo 8: Presenta la documentación consultada durante el desarrollo del proyecto.

Capítulo 2

Tecnologías y plataformas

En este capítulo ilustraremos las múltiples plataformas y tecnologías utilizadas durante el desarrollo e implementación del sistema asignado a este proyecto fin de carrera. Dichas plataformas se dividirán en función de si se han utilizado durante el desarrollo para la parte de dispositivos móviles o para alguno de los servidores. Finalmente se explicaran en detalle los entornos de desarrollo empleados

2.1 Dispositivos móviles

Android

Android es un sistema operativo libre, gratuito y multiplataforma basado en el kernel de Linux. Cualquier desarrollador puede crear aplicaciones para Android sin la necesidad de pagar membresías anuales para obtener el kit de desarrollo (SDK). utiliza una variación del lenguaje de programación Java que es diferente a la Java ME[9]. En la Figura 5 se muestra el diagrama de la arquitectura de Android.

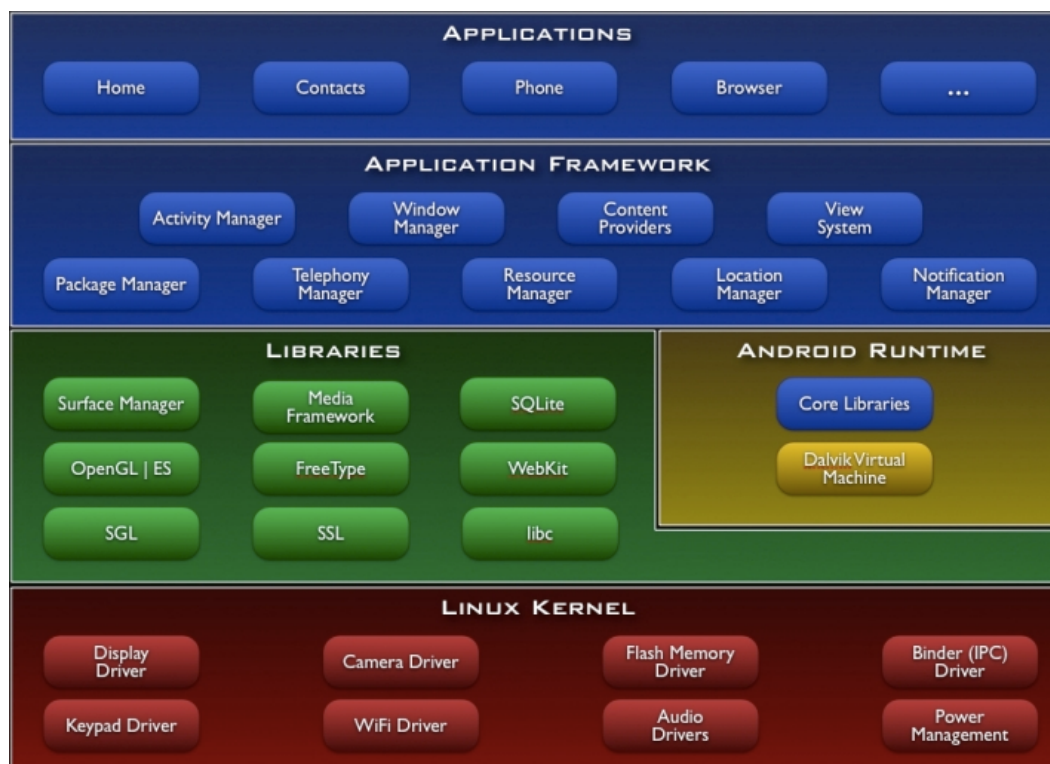


Figura 5. Arquitectura interna Android[10].

Para una mayor claridad, vamos a explicar a continuación cada una de las capas expuestas en la figura anterior, empezando por la capa inferior.

- **Kernel Linux:** El núcleo actúa como una capa de abstracción entre el hardware y el resto de la pila de software. Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El desarrollador no accede directamente a esta capa, sino que debe utilizar las librerías disponibles en capas superiores[11].
- **Librerías:** La siguiente capa, situada justo encima del kernel, esta compuesta de las librerías nativas de Android. Están escritas en C o C++ y compiladas para la arquitectura hardware específica de cada dispositivo[11]. El objetivo de estas librerías es proporcionar funcionalidad a las aplicaciones para tareas que se repiten con frecuencia, evitando tener que codificarlas cada vez y garantizando que se llevan a cabo de forma eficiente.
- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones del lenguaje Java. El componente principal del entorno de ejecución de Android es la máquina virtual Dalvik. Las aplicaciones se codifican en Java y son compiladas en un formato específico para que esta máquina virtual las ejecute. La ventaja de esto es que las aplicaciones se compilan una única vez, asegurando de esta forma que al ser distribuidas, podrán ejecutarse en cualquier dispositivo Android que disponga de la versión mínima del sistema operativo que requiera la aplicación.

Es necesario aclarar que Dalvik es una variación de la máquina virtual de Java. De esta forma, Java solo es utilizado como lenguaje de programación. Los ejecutables que se generan con el SDK de Android tienen la extensión .dex, que es específico para Dalvik[11].

- **Entorno de trabajo de aplicaciones:** La siguiente capa esta formada por todas las clases y servicios que utilizan directamente las aplicaciones para realizar sus funciones. La mayoría de los componentes de esta capa son librerías Java que acceden a los recursos de las capas anteriores a través de la máquina virtual Dalvik[11].
- **Aplicaciones:** En esta última capa se incluyen todas las aplicaciones del dispositivo, tanto las que tienen interfaz de usuario como las que no, las nativas y las administradas, las que vienen preinstaladas y las que ha instalado el usuario[11].

A la hora de afrontar el desarrollo de software para este sistema operativo, se tiene a disposición un entorno sumamente poderoso, resultando accesible cualquier parte del sistema operativo. El potencial de este sistema operativo se basa en la libertad que nos ofrece a la hora de programar.

Para empezar con el desarrollo de una aplicación para Android necesitaremos descargar un entorno de aplicación y desarrollo, el SDK.

Ciclo de vida aplicación Android

El primer paso para explicar el ciclo de vida de una aplicación en Android es comprender que es una Activity, ya que estas son las clases en las que se basa el funcionamiento de las aplicaciones.

Una Activity es una clase única, enfocada en lo que el usuario puede hacer. Casi todas las actividades interactúan con el usuario, de forma que crean ventanas con una interfaz con la que poder interactuar[10].

El ciclo de vida de una aplicación en Android, asociado al de las Activities que la componen, es manejado por el sistema operativo, basándose en las necesidades del usuario, los recursos disponibles, etc. Si tenemos una aplicación que está consumiendo muchos recursos y arrancamos otra nueva aplicación, el sistema operativo probablemente le diga a la aplicación que se queda en segundo plano que libere todo lo que pueda, y si es necesario la cerrará. En Android los recursos son normalmente muy limitados y por eso el sistema operativo tiene más control sobre las aplicaciones que en programas de escritorio.

En la mayoría de los casos, cada aplicación Android corre en su propio proceso de Linux. Este proceso es creado para la aplicación cuando la arrancamos y seguirá corriendo hasta que no sea necesario y el sistema reclame recursos para otras aplicaciones y se los dé a éstas[12].

Android puede en cualquier momento pausar, parar o destruir nuestra aplicación según las necesidades; de nosotros depende controlar todos estos eventos para hacer una aplicación robusta.

En la Figura 6, proporcionada por Google, podemos apreciar como seria el ciclo de vida de una aplicación.

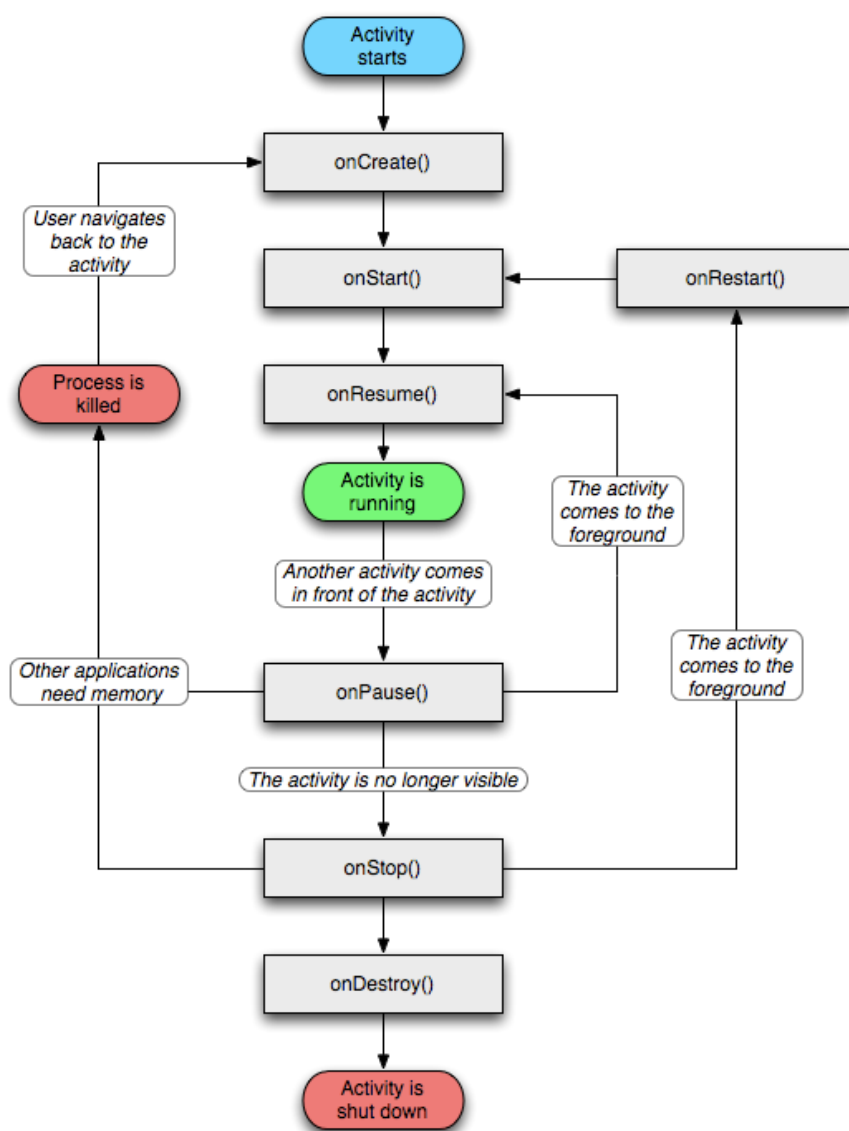


Figura 6. Ciclo de vida aplicación Android[10].

Para poder entender el funcionamiento del diagrama expuesto anteriormente, es necesario explicar en que consisten cada uno de los métodos que aparecen en él.

- **onCreate():** Se dispara cuando la Activity es llamada por primera vez, inicializando la aplicación y creando las vistas. Este método da acceso al estado en el que estaba la aplicación cuando se cerró. La siguiente llamada siempre será a onStart()[10].
- **onRestart():** Se ejecuta cuando la Activity ha sido parada, y quiere volver a utilizarse. Esta llamada se realiza tras onStop() y posteriormente se llama a onStart()[10].
- **onStart():** Se llama cuando la Activity se vuelve visible para el usuario. Seguido por onResume() si la Activity pasa a primer plano u onStop() si pasa a estar oculta[10].

- **onResume():** Se ejecuta cuando el usuario empieza a interactuar con la aplicación. Cuando el usuario termina de utilizarla se llama a onPause()[10].
- **onPause():** Se ejecuta cuando el sistema arranca una nueva Activity que necesita los recursos del sistema. Esta llamada debe ser rápida ya que hasta que no finalice su ejecución no se podrá lanzar una nueva Activity. Seguida por onResume() si la Activity vuelve al primer plano u onStop() si se vuelve invisible para el usuario[10].
- **onStop():** Se ejecuta cuando la Activity ya no es visible para el usuario porque otra ha pasado a primer plano. Tras esta llamada puede ejecutarse onStart() para que la Activity vuelva a aparecer en primer plano u onDestroy() para cerrar la aplicación. En este punto también es posible que el sistema elimine el proceso, porque otros necesiten los recursos[10].
- **onDestroy():** Es la llamada final de la Activity, tras la cual queda totalmente destruida. Puede producirse por necesidad de recursos del sistema o provocada por el usuario[10].

Para decidir que proceso debe ser liquidado, Android ordena los procesos por importancia, según la siguiente jerarquía[13]:

- **Proceso en primer plano:** Es la aplicación que contiene la actividad que ahora mismo se está mostrando en pantalla (Se ha llamado al método onResume()). Habrá muy pocos procesos primarios corriendo a la vez en el sistema y estos procesos solamente se liquidarán si la memoria es tan baja que ni matando al resto de procesos tenemos los recursos necesarios.
- **Proceso visible:** Es el que contiene una actividad que es visible, pero no en primera fila (se ha llamado al método onPause()). Estos procesos son considerados importantes por el sistema operativo y normalmente no se cierran.
- **Proceso de servicio:** Es un servicio como los de cualquier Unix. Estos procesos hacen cosas en segundo plano que normalmente son importantes, el sistema nunca va a liquidar un servicio a menos que sea necesario para mantener vivos todos los visibles y de primer plano.
- **Proceso en segundo plano:** Es un proceso que contiene una actividad que actualmente no es visible por el usuario, estos procesos no tienen demasiada importancia, puede ser un programa iniciado hace tiempo y que no se ha vuelto a usar; pasa a estar en segundo plano. Por eso es importante cuando nuestra aplicación pase a segundo plano, liberar en la medida de lo posible todos los recursos que podamos.
- **Proceso vacío:** Es un proceso que ya no alberga nada, lo usa Android como caché para cuando se crea un proceso nuevo.

2.2 Servidor Whatscine

Java

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems a principios de los años 90[14]. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo. Java es, en Septiembre de 2013, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de webs[15].

Web service

Un servicio web (en inglés, Web Service) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software, desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares. Un servicio web atiende las peticiones de los clientes y les envía los recursos solicitados[16].

Los estándares empleados son los siguientes:

- **Web Services Protocol Stack:** Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- **XML:** Es el formato estándar para los datos que se vayan a intercambiar.
- **SOAP o XML-RPC:** Protocolos sobre los que se establece el intercambio.
- **Otros protocolos:** los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP, FTP o SMTP.
- **WSDL:** Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- **UDDI:** Protocolo para publicar la información de los servicios Web. Permite comprobar qué servicios web están disponibles.

- **WS-Security:** Protocolo de seguridad aceptado como estándar por OASIS. Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

Los servicios web aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen. Además fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento. Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

JSP

JSP es una tecnología que permite, a los desarrolladores y diseñadores Web, un fácil y rápido desarrollo y mantenimiento de páginas web dinámicas y ricas en información. Como parte de la familia de tecnologías Java, JSP permite un rápido desarrollo de aplicaciones basadas en web, independientemente de su plataforma. Esta tecnología separa la interfaz de usuario de la generación de contenido, dando la opción a los diseñadores de desarrollar la vista de la página sin necesidad de alterar su contenido dinámico.

La tecnología JSP utiliza etiquetas XML para encapsular la lógica que genera el contenido de la página. La lógica de la aplicación puede residir en los recursos basados en servidor, a los que la página accede con estas etiquetas. Todas las etiquetas con formato HTML o XML, se pasan directamente a la página de respuesta. Al separar la lógica de la página de su diseño y presentación, y dando soporte a componentes reutilizables, JSP hace que sea mas rápido y fácil construir aplicaciones basadas en web[17].

Para desplegar y correr JSP, es necesario un servidor web compatible con contenedores servlet como Apache Tomcat o Jetty.

El rendimiento de una página JSP es el mismo que tendría el servlet equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

HTML

HTML hace referencia al lenguaje de marcado predominante para la elaboración de páginas web. Básicamente se trata de un conjunto de etiquetas que sirve para definir el texto y otros elementos que compondrán una pagina web[18]. Se utiliza para describir y traducir la estructura y la información en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>), en un documento de texto. También puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir o hacer referencia a un tipo de programa llamado script, el cual puede afectar al comportamiento de navegadores web y otros procesadores de HTML. HTML consta de varios componentes vitales, entre ellos los elementos y sus atributos, tipos de datos y la declaración de tipo de documento[19].

CSS

CSS es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura.

CSS se utiliza para dar estilo a documentos HTML y XML, separando el contenido de la presentación. Los Estilos definen la forma de mostrar los elementos HTML y XML. CSS permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo. Cualquier cambio en el estilo marcado para un elemento en la CSS afectará a todas las páginas vinculadas a esa CSS en las que aparezca ese elemento.

El funcionamiento de CSS se basa en reglas, es decir, declaraciones sobre el estilo de uno o más elementos. Las hojas de estilo están compuestas por una o más de esas reglas aplicadas a un documento HTML o XML. La regla tiene dos partes: un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne[20].

MySQL

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

MySQL es un sistema de gestión de bases de datos relacional, multihilo, multiusuario y de licencia libre GPL. Ha sido desarrollado por Oracle y cuenta con gran prestigio a nivel mundial.

MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, esto hace que MySQL sea ideal para este tipo de aplicaciones[21].

Apache Tomcat

Apache Tomcat es una implementación software de código abierto para servlet Java y tecnologías JSP. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAs. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de tráfico y alta disponibilidad.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java[22].

2.3 Servidor sala

C

C es un lenguaje de programación orientado a la implementación de sistemas operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

Se trata de un lenguaje de tipos de datos estáticos, débilmente tipificado, de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos[23].

Al igual que ocurre con Java, C es uno de los lenguajes de programación mas utilizados actualmente[15].

2.4 Entornos de desarrollo

A continuación voy a explicar con mayor detalle los distintos entornos de desarrollo utilizados, para qué partes del proyecto ha sido necesario cada uno de ellos y cual ha sido el motivo de su elección.

NetBeans IDE

NetBeans es un entorno de desarrollo integrado, hecho principalmente para el lenguaje de programación Java. Es un producto libre y gratuito, sin restricciones de uso. La plataforma permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un modulo es una archivo Java que contiene clases escritas para interactuar con las APIs de NetBeans. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos[24].

Durante el desarrollo del proyecto se tomo la decisión de utilizar Netbeans como entorno de desarrollo del servidor propio de Whatscine. El servidor se basa en un servicio web implementado en Java, lo que hacía de NetBeans un entorno ideal, además, facilitaba mucho la interacción con la base de datos implementada en MySQL y el desarrollo de la pagina web en JSP. Otro de los puntos fuertes de NetBeans era la facilidad proporcionada para integrar el código desarrollado en un servidor Apache Tomcat.

Eclipse

Eclipse es un entorno multi-lenguaje de desarrollo integrado, escrito principalmente en Java, que comprende un área de trabajo base y un sistema extensible de módulos para personalizar el entorno. Se puede utilizar para el desarrollo de aplicaciones Java u, por medio de los distintos módulos, otros lenguajes de programación[25].

En nuestro caso, Eclipse se ha utilizado para el desarrollo de la parte del cliente Android. Se decidió utilizar este entorno ya que es el utilizado por defecto para el desarrollo de aplicaciones Android. Para tal fin basta con instalar un modulo específico. Esta herramienta, llamada ADT, facilita enormemente la creación de proyectos, su implementación, depuración y ejecución, siendo de esta forma altamente recomendable su uso.

Xcode

Xcode es el entorno de desarrollo integrado de Apple Inc. y se suministra gratuitamente junto con Mac OS X.

Xcode incluye la colección de compiladores del proyecto GNU (GCC), y puede compilar código C, C++, Objective-C, Objective-C++, Java y AppleScript mediante una amplia gama de modelos de programación, incluyendo, pero no limitado a Cocoa, Carbón y Java[26].

El desarrollo del módulo perteneciente a este proyecto, que se incluirá en el servidor propio de cada sala de cine, está implementado en lenguaje C. Esto se debe a que es en este lenguaje en el que se encontraba el servidor ya implementado, antes del comienzo de este proyecto. Como Xcode incluye de forma nativa el compilador de C, se utilizó este entorno de desarrollo, en concordancia con la decisión de utilizar un entorno distinto, siempre que fuera posible, para cada parte independiente del proyecto.

Capítulo 3

Funcionalidad y tareas del sistema

En este capítulo se mostrarán cuales son las distintas funcionalidades y tareas que realiza el sistema desarrollado en el proyecto.

El sistema realiza cuatro tareas generales, las cuales se llevan a cabo gracias a la interacción entre los distintos componentes. Estas tareas engloban un mayor número de subtareas, encargadas de diversas funciones más concretas, y generalmente realizadas por un solo componente del sistema.

3.1 Tarea 1: Registro actividad usuarios

Esta tarea consiste en registrar la actividad de los usuarios dentro del sistema Whatscine. Para poder entender mejor en que consiste, el primer paso es mostrar cuales son los actores y componentes dentro del que sería un escenario de uso real.

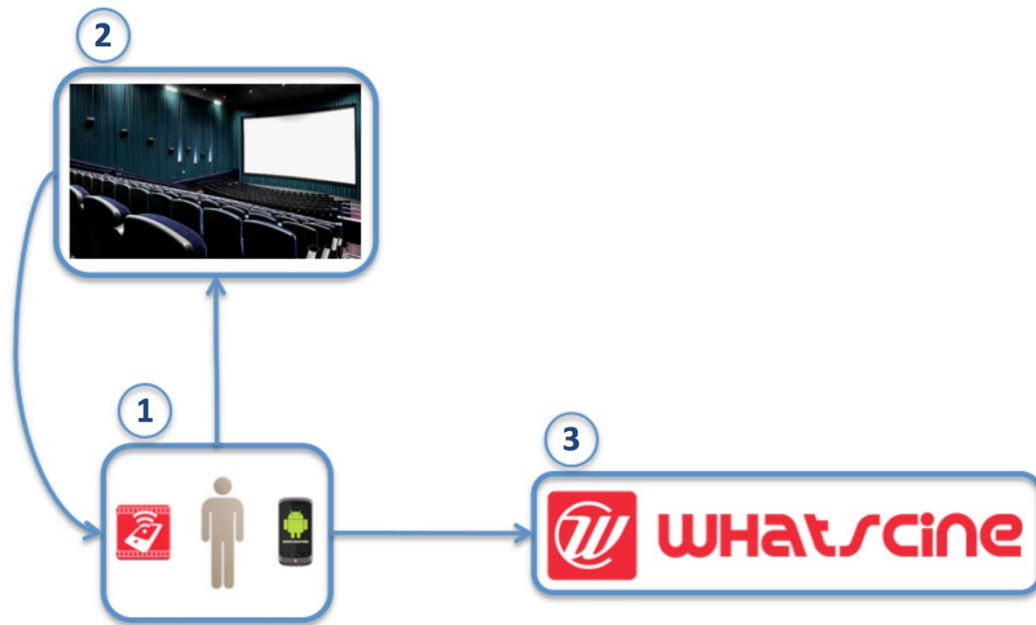


Figura 7. Escenario tarea 1.

En la Figura 7 puede verse que son 3 los elementos que intervienen en esta tarea:

1. Espectador y su dispositivo Android.
2. Sala de cine.
3. Sistema Whatscine.

A continuación se explicará en detalle como se realiza esta tarea, siendo el de la Figura 8 su diagrama de flujo.

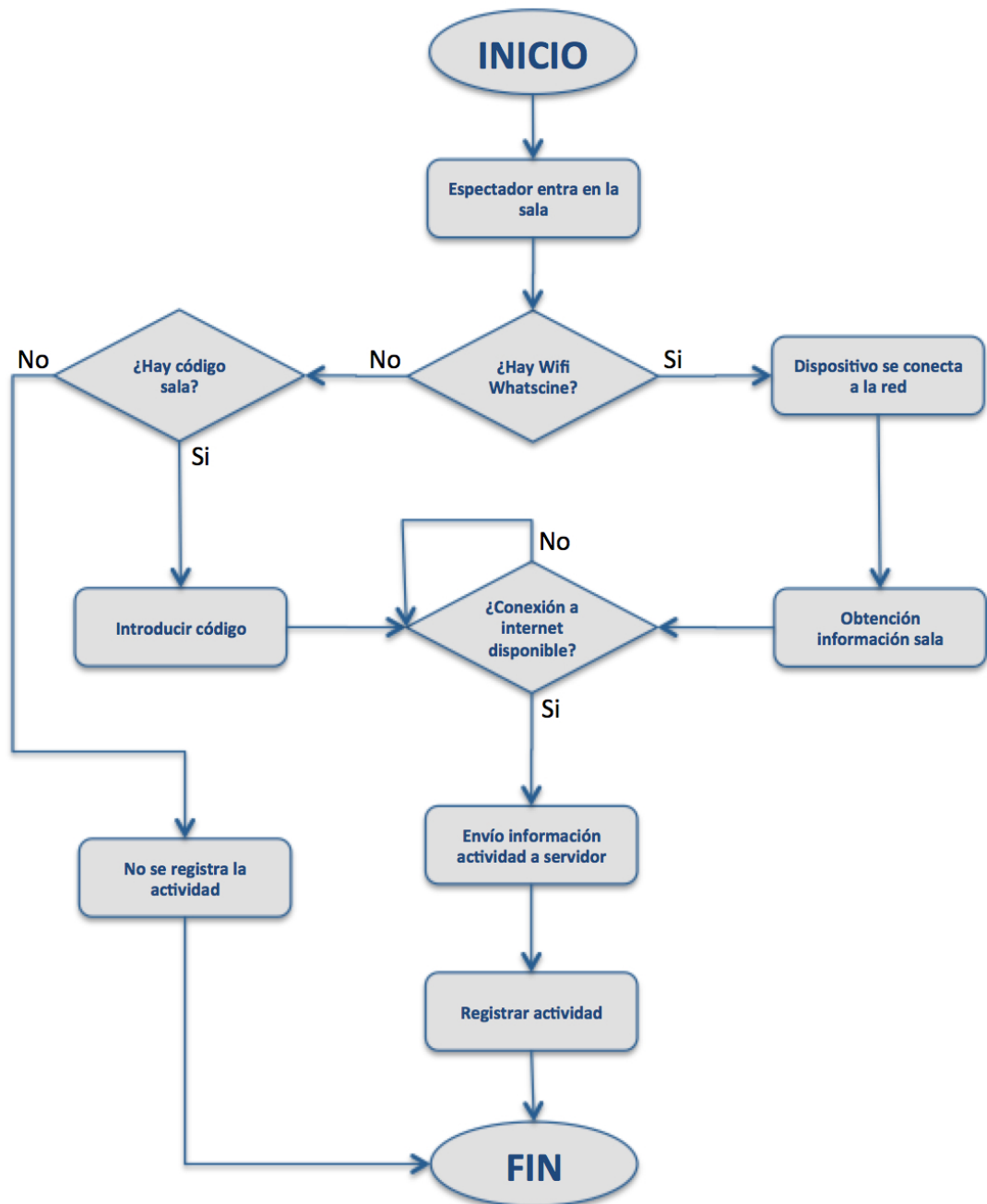


Figura 8. Diagrama de flujo tarea 1.

El registro de la actividad de los usuarios se lleva a cabo mediante el siguiente proceso:

1. El usuario accede a una sala de cine asociada al sistema Whatscine. Llevará consigo un dispositivo Android con la aplicación instalada.
2. En este punto del proceso se pueden dar dos casos distintos, que la sala tenga una red wifi del sistema Whatscine o que no.
 - a. Si la sala no tiene instalada una red inalámbrica para el sistema, pero sí está asociada, se facilitará un código de identificación en la pantalla de la misma. En este caso la aplicación solicitará dicho código. En el caso

- de que no se dispusiera del código, o no se facilitara a la aplicación la actividad no quedaría registrada.
- b. En el caso de que la sala disponga de red inalámbrica, el usuario tendrá que conectar su dispositivo móvil. La aplicación, de manera automática, obtendrá la información necesaria de la sala para realizar el registro.
 3. La aplicación preparará la información necesaria para el registro, tanto de la sala como del dispositivo, y quedará a la espera de tener conexión a internet para enviarla al servidor de Whatscine.
 4. Se envía la información relativa a la actividad.
 5. El servidor almacenará la información recibida de forma permanente. Llegados a este punto, el registro se habrá realizado, finalizando así la tarea.

Para realizar el registro hacen falta los datos mostrados en la Tabla 4.

SALA	MAC router	Con wifi
	SSID red	Con wifi
	Código	Sin wifi
DISPOSITIVO	Número de serie	Siempre
	Fecha	Siempre
	Hora	Siempre

Tabla 4. Datos necesarios para el registro.

Los tres primeros datos son los que identificarán la sala donde se encuentra el usuario, tanto en el caso que haya red wifi como si no. Los tres últimos se obtienen del dispositivo móvil. El número de serie nos sirve como identificador único de cada dispositivo, y la fecha y hora para saber en que momento se realizó la actividad.

3.2 Tarea 2: Usuario premiado

Esta tarea se basa en la gestión de los premios obtenidos en los juegos de Whatscine, desde que se generan en la sala de cine, hasta que quedan registrados y disponibles para su uso. Para poder entender mejor en que consiste, el primer paso es mostrar cuales son los actores y componentes dentro del que seria un escenario de uso real.



Figura 9. Escenario tarea 2.

En la Figura 9 puede verse que son 3 los elementos que intervienen en esta tarea:

1. Sala de cine.
2. Espectador y su dispositivo Android.
3. Sistema Whatscine.

A continuación, en la Figura 10, se muestra el diagrama de flujo de la tarea.

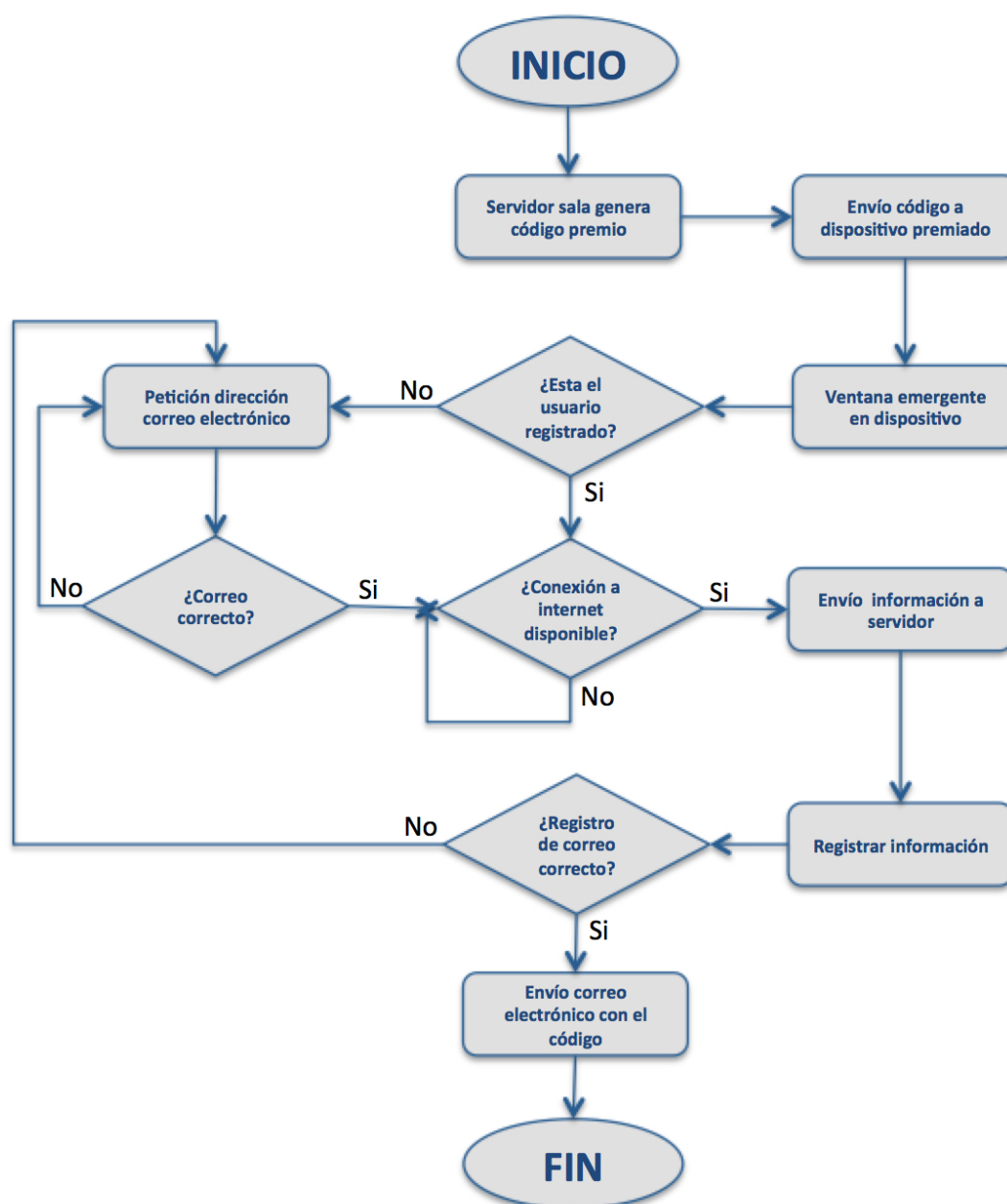


Figura 10. Diagrama de flujo tarea 2.

Como puede verse, la tarea consta de varios pasos.

1. Tras finalizar un juego, el sistema integrado en la sala correspondiente genera un código de premio. Este código se envía mediante la conexión Wifi al dispositivo ganador.
2. El dispositivo, al recibir el código, muestra una ventana avisando al usuario de que ha resultado ganador.
3. Llegados a este punto, es imprescindible que el usuario se registre en el sistema Whatscine proporcionando un correo electrónico. Si el usuario ya se había registrado anteriormente, el proceso continuará. Por el contrario, si no estaba registrado, la aplicación solicitará un correo electrónico comprobando que la dirección facilitada sea correcta, volviendo a solicitarla si no lo es.

4. Cuando el dispositivo obtiene conexión a internet, la aplicación envía el código del premio y el correo electrónico al servidor.
5. El servidor, al recibir la información, procede a almacenar la información relativa al premio, asociándola a la actividad en la que se obtuvo. Además procederá al registro del usuario que ha obtenido el premio. En este punto podría darse la situación de que el registro no fuera posible debido a que el correo no fuera válido. Si esto sucediera, el servidor solicitaría a la aplicación una nueva dirección de correo electrónico, volviendo así al paso 3 de esta tarea.
6. Una vez el registro del premio y del usuario han sido correctos, el servidor enviará un correo electrónico a la dirección facilitada, con el código y la información necesaria para obtener su premio.

El código generado por el servidor de la sala está compuesto por 8 caracteres imprimibles ASCII. Se crea mediante un algoritmo que se encarga de codificar información relevante para el sistema Whatscine.

3.3 Tarea 3: Canjear código premiado

Esta tarea es la encargada de controlar la entrega de premios en el sistema Whatscine.

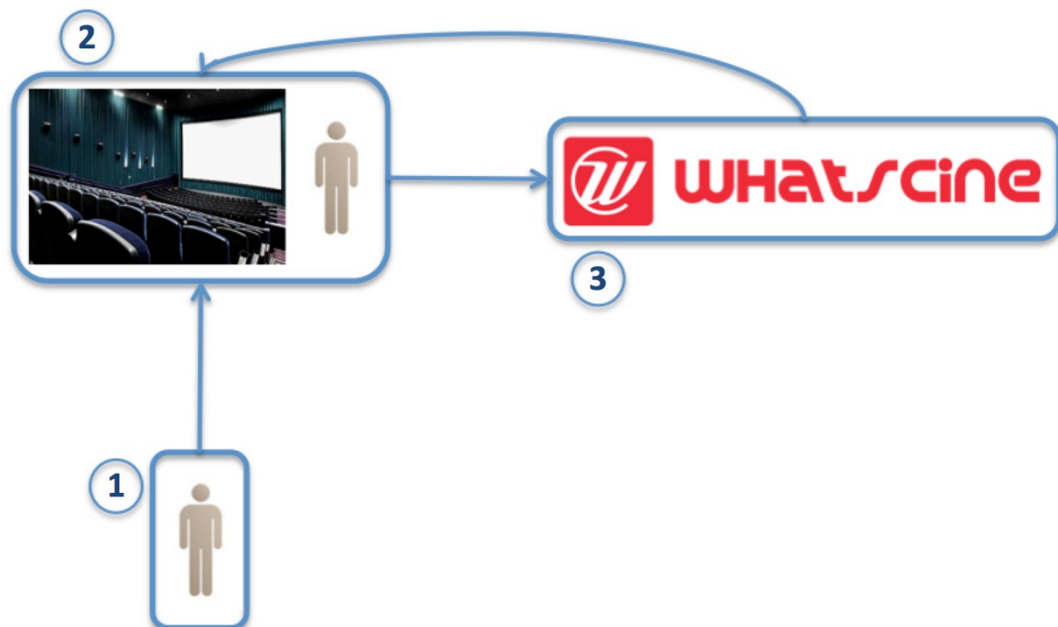


Figura 11. Escenario tarea 3.

CAPÍTULO 3: Funcionalidad y tareas del sistema

En la Figura 11 puede verse que son 3 los elementos que intervienen en esta tarea:

1. Espectador.
2. Personal del cine.
3. Sistema Whatscine.

En la Figura 12 se muestra el diagrama de flujo que representa esta tarea.

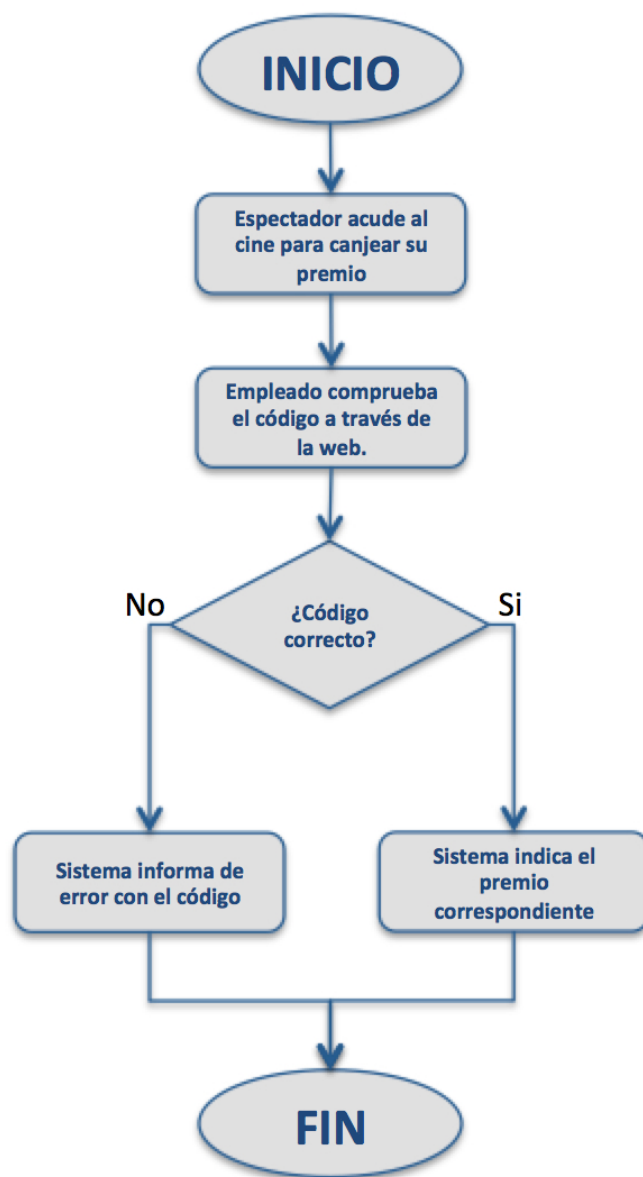


Figura 12. Diagrama de flujo tarea 3.

El proceso necesario para realizar el canje de un código premiado es el que se muestra a continuación:

1. Una vez que el ganador de algún juego en el sistema Whatscine, recibe un correo electrónico con el código del premio, puede acudir al cine a canjearlo.
2. El personal del cine le solicitará el código y procederá a comprobar su validez por medio de la web de Whatscine.
3. El servidor realizará una serie de comprobaciones con el código proporcionado y mostrará un mensaje indicando si es correcto o no.
 - a. Si el código es correcto, el sistema mostrará cual es el premio correspondiente, marcando el código como utilizado.
 - b. Si el código es incorrecto, el sistema mostrara un mensaje indicando cual ha sido el problema.

Son varios los motivos por los que el código puede no ser correcto:

- Ha sido utilizado previamente. Este mensaje se mostrara cuando el código aparezca marcado como ya utilizado en el sistema.
- No se encuentra ningún premio en el sistema asociado a dicho código. Esto ocurrirá cuando el código no aparezca registrado entre los premiados, pero sin embargo esté correctamente formado.
- El código no es correcto. Este caso ocurrirá cuando el código no esté correctamente formado.

3.4 Tarea 4: Gestión salas de cine

Esta tarea está formada por un grupo de sub-tareas que sirven para gestionar la información de las salas asociadas al sistema Whatscine.

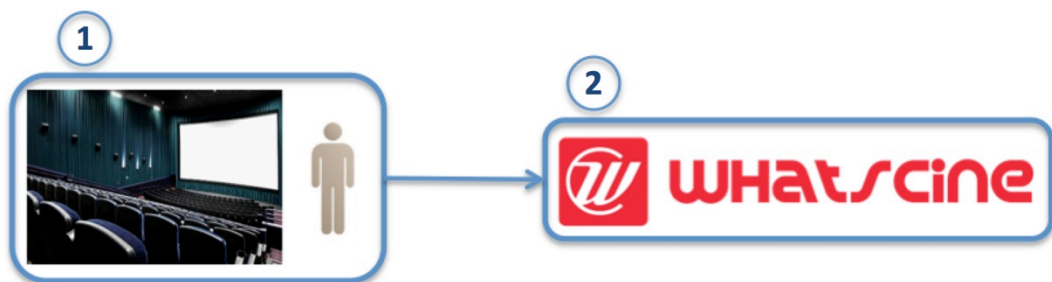


Figura 13. Escenario tarea 4.

En la Figura 13 puede verse que son 2 los elementos que intervienen en esta tarea:

1. Personal del cine.
2. Sistema Whatscine.

El conjunto de las sub-tareas puede verse en la Figura 14.



Figura 14. Diagrama tarea 4.

La gestión de la información disponible de las salas de cine asociadas al sistema Whatscine, se realiza a través de una página web, desarrollada dentro de este proyecto. Para poder realizar dicha gestión es necesario tener una cuenta registrada en el sistema, con nombre de usuario y contraseña, de modo que el acceso este limitado. Una vez se accede a dicha página, son 3 las opciones que se disponen.

1. Insertar sala: Servirá para introducir una nueva sala de cine en el sistema.
2. Modificar sala: Permite cambiar alguno de los datos de las salas asociadas.
3. Eliminar sala: Podemos borrar una sala concreta del sistema.

La información que se puede almacenar de cada sala de cine, y que sirve para su identificación es:

- MAC router.
- SSID red
- Código sala.
- Número de sala real en el cine.

Al realizar el registro de una nueva sala, gracias al sistema de login, esta queda asociada al cine al que pertenece.

Capítulo 4

Análisis del sistema

En este capítulo se realizará un análisis del sistema, centrado en su implementación técnica. En primer lugar se expondrá una visión general del sistema completo, para posteriormente explicar, con mayor detalle, cada uno de los diferentes componentes.

4.1 Arquitectura general

En la Figura 15 se muestra la arquitectura general del sistema.

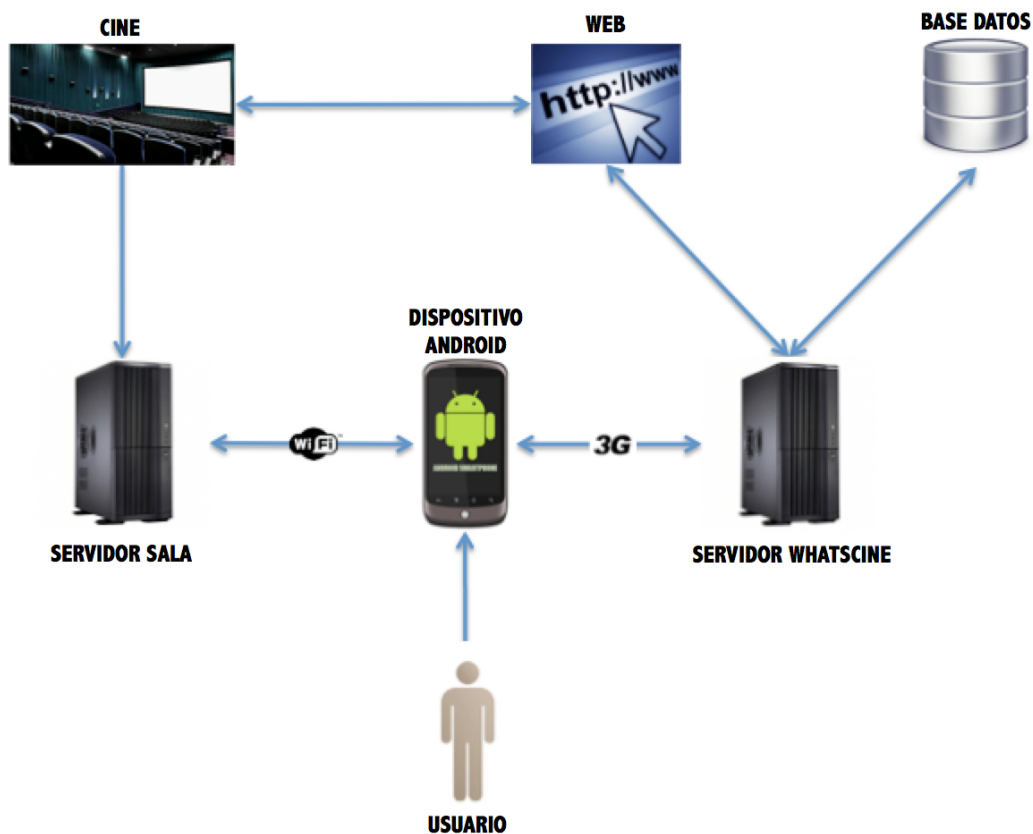


Figura 15. Arquitectura general del sistema completo.

Como puede apreciarse, el sistema se compone de tres partes diferenciadas, el servidor ubicado en cada sala de cine, el servidor propio de Whatscine y el dispositivo Android de cada usuario. Para el desarrollo de este proyecto ha sido necesario implementar distintos módulos que se integran dentro de estas tres partes. En el diagrama expuesto, también se muestra que hay dos clases de usuarios que hacen uso del sistema:

- Espectadores: Hacen uso de las funciones del sistema mediante sus dispositivos móviles.
- Personal cine: Utilizan la web de Whatscine para gestionar la información de los premios y las salas, para el correcto funcionamiento del sistema.

A continuación vamos a ver con mayor detalle cada componente del sistema de forma independiente.

4.2 Servidor Whatscine

La mayor parte de la funcionalidad del proyecto recae sobre este servidor.

A diferencia de las otras dos partes, el servidor propio de Whatscine ha sido desarrollado desde cero y por completo durante el desarrollo del proyecto. Es por esto que se trata del componente más complejo y que mayor trabajo requirió en su implementación.

El servidor se compone de los bloques que pueden verse en la Figura 16.

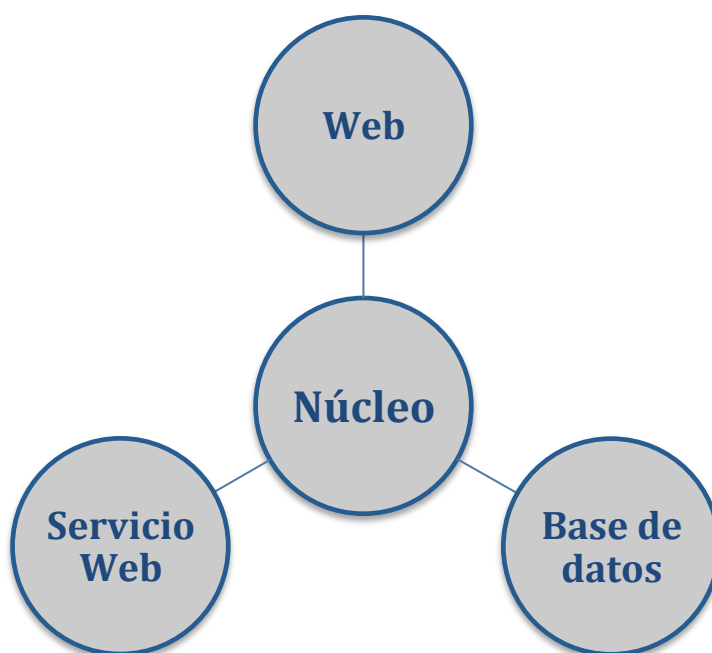


Figura 16. Bloques servidor Whatscine.

Servicio Web

Se trata de la vía de comunicación de la aplicación, instalada en los dispositivos móviles, con el servidor. Para que el cliente pueda tener acceso al servicio, este tiene que ser publicado. La dirección de nuestro servicio web, siendo la dirección ip la del equipo donde finalmente se monte el servidor, es la siguiente:

`"http://dirección.ip:8080/WSCine/WebServiceCine"`

El servicio web se compone de tres métodos para el intercambio de información.

Conexión3G

Recibe la información del dispositivo móvil cuando se da el caso de que la sala no tiene red wifi instalada y devuelve un mensaje de respuesta.

Los parámetros que se intercambian con este método se pueden ver en la Tabla 5.

	Parámetro	Tipo
Entrada	código	String
	id	String
	hora	String
	fecha	String
	gan	String
	mail	String
	cGanador	String
Salida	respuesta	String

Tabla 5. Parámetros método conexión3G.

Los parámetros de entrada: código, id, fecha y hora sirven para realizar el registro de la actividad del usuario (tarea 1), mientras que los parámetros: gan, mail y cGanador permiten gestionar la información necesaria si el usuario resulta premiado (tarea 2). El parámetro de salida “respuesta” transmite a la aplicación Android el resultado de la operación realizada en el servidor. Avisará tanto si todo ha ido bien, como en el caso de que se haya producido algún error, incluso solicitando el reenvío de algún dato erróneo.

ConexiónWIFI

Este método recibe la información del dispositivo móvil cuando en la sala de cine hay una red wifi instalada, como en el caso anterior, devuelve un mensaje con la respuesta.

Los parámetros intercambiados son los que aparecen en la Tabla 6.

	Parámetro	Tipo
Entrada	SSIDap	String
	MACap	String
	id	String
	hora	String
	fecha	String
	gan	String
	mail	String
	cGanador	String
Salida	respuesta	String

Tabla 6. Parámetros método conexiónWIFI.

Los parámetros de entrada: SSIDap, MACap, id, fecha y hora sirven para realizar el registro de la actividad del usuario (tarea 1), mientras que los parámetros: gan, mail y cGanador permiten gestionar la información necesaria si el usuario resulta premiado (tarea 2).

El parámetro de salida “respuesta” transmite a la aplicación Android el resultado de la operación realizada en el servidor. Avisará tanto si todo ha ido bien, como en el

caso de que se haya producido algún error, incluso solicitando el reenvío de algún dato erróneo.

Correo

Este método es el encargado de recibir la información del premio cuando se ha requerido el reenvío del correo electrónico. Si la primera vez que se intenta realizar el registro del usuario el servidor rechaza el correo proporcionado, se solicitará uno nuevo, siendo este el método que se encargue de recibir la respuesta hasta obtener uno válido.

Los parámetros intercambiados son los que aparecen en la Tabla 7.

	Parámetro	Tipo
Entrada	id	String
	hora	String
	fecha	String
	mail	String
	cGanador	String
Salida	respuesta	String

Tabla 7. Parámetros método correo.

Todos los parámetros de entrada sirven para gestionar la información necesaria si el usuario resulta premiado (tarea 2).

El servicio Web solo funciona como vía de comunicación con el dispositivo móvil, en él no se realiza ninguna función adicional. Toda la carga de control y gestión de la información se realiza en el núcleo.

Página Web

La página web permite la gestión y el control de la información de las salas de cine. Cada cine tiene la posibilidad de registrarse en el sistema a través de esta página, creando una cuenta de usuario, con nombre y contraseña. Con esta cuenta puede iniciar sesión y gestionar la información relativa a las salas que quiera asociar a Whatscine. Además, también se proporciona el control de los premios.

La página web se compone de cinco apartados. En la Figura 17 puede verse su arquitectura general.

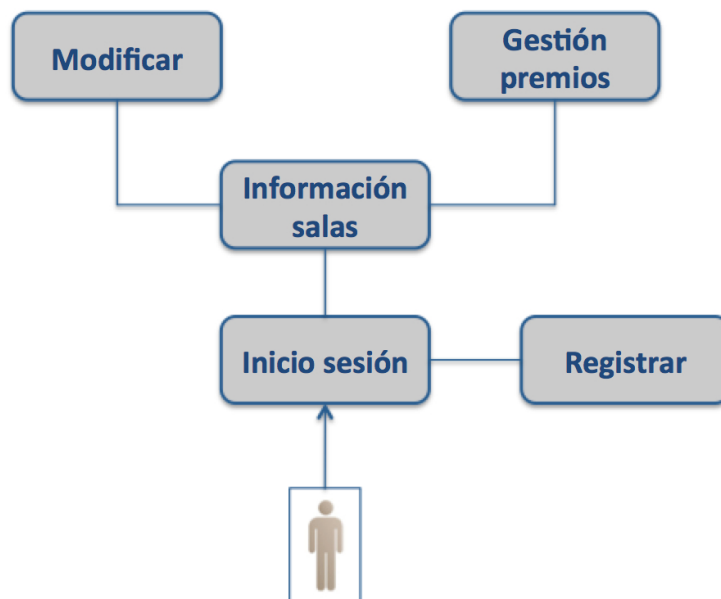


Figura 17. Arquitectura general página web.

Página inicio de sesión

Esta página es el punto de acceso para los usuarios al resto del sistema web. Desde aquí se podrá acceder a la información de cada cine mediante un sistema de inicio de sesión. Además, también se tendrá acceso a la página de registro de usuario por medio del botón “Registro”.

La vista de esta página puede verse en la Figura 18.

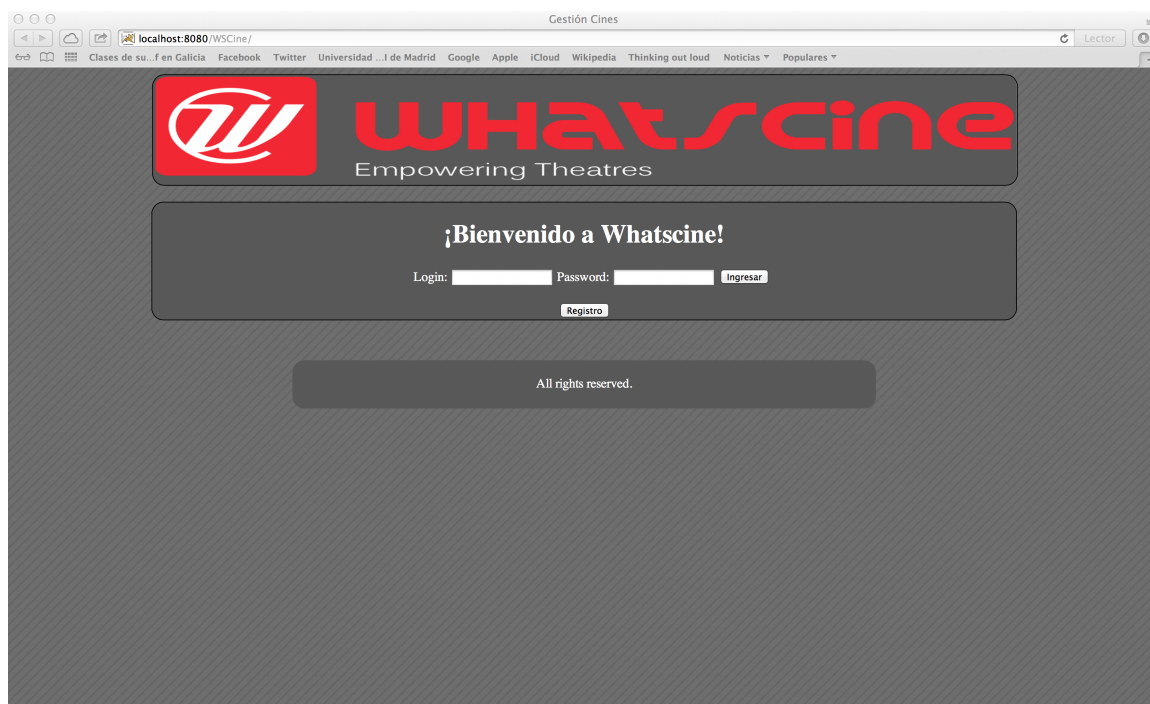


Figura 18. Vista página inicio de sesión.

Página registro de usuario

A través de esta página, los cines que no tengan creada una cuenta de acceso al sistema Whatscine podrán registrarse. Se dispone también de un botón “Volver” que nos redirecciona de nuevo a la página de inicio de sesión.

La vista de la página de registro puede verse en la Figura 19.

Figura 19. Vista página registro de usuario.

Página información salas de cine

Esta página nos muestra toda la información de las salas asociadas al sistema Whatscine, almacenada en la base de datos.

Se compone de tres bloques:

- Tabla con todas las salas del cine que ha iniciado sesión y su información. Además se da la opción de eliminar o modificar cada una de ellas.
- Formulario para insertar nuevas salas.
- Un botón para salir del sistema y otro para gestionar los códigos de premio.

La Figura 20 muestra el aspecto de esta página.

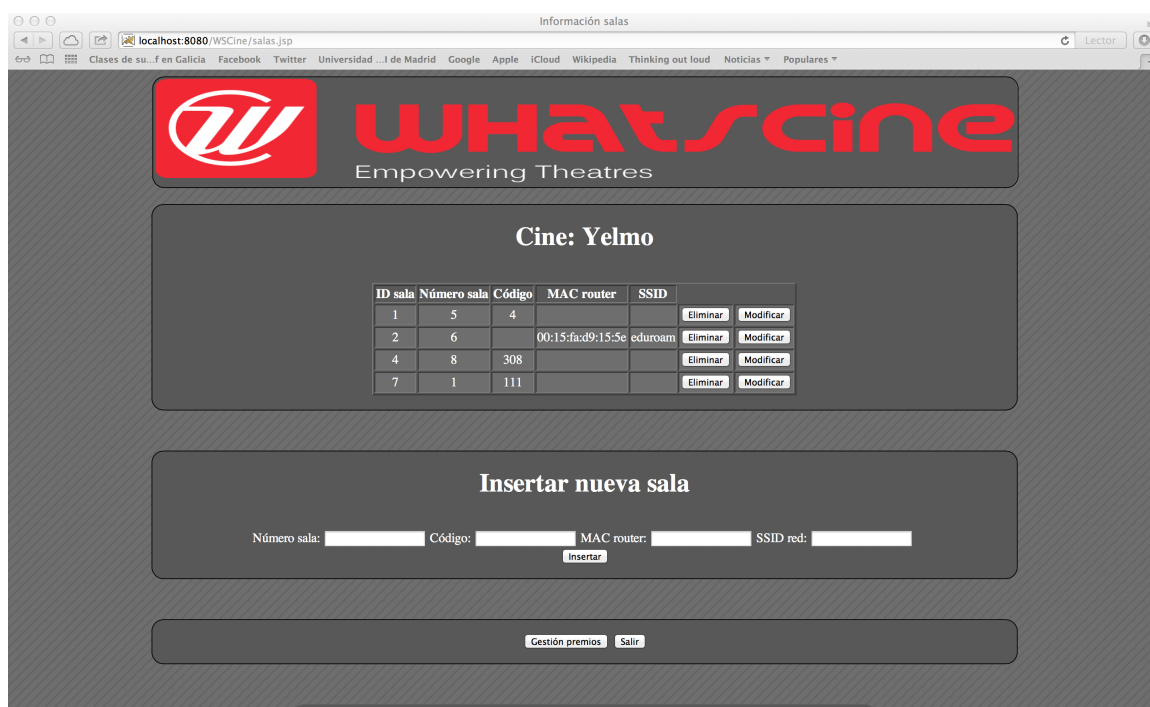


Figura 20. Vista página información salas de cine.

Página modificar

Presenta un formulario con los campos disponibles para modificar de una sala concreta. La información ya disponible previamente aparece como valor por defecto. Además dispone de un botón para volver a la página de información. Son tres los campos que pueden modificarse de una sala: código, MAC y SSID. En la Figura 21 se puede apreciar el aspecto de esta página.



Figura 21. Vista página modificar.

Página gestión de premios

Proporciona el control sobre los premios dentro del sistema. Te indica el estado de un código que introduzcas a través de un formulario. Como en la página anterior, incluye un botón para volver a la página de información.

La vista de la página de gestión de premios puede verse en la Figura 22.



Figura 22. Vista página gestión premios.

Base de datos

Se encarga de almacenar, de forma persistente, toda la información necesaria para el sistema. La base de datos implementada en el servidor está desarrollada en MySQL. El primer paso que se llevó a cabo fue realizar un diseño para la base de datos que se ajustara a las características y necesidades del sistema. Para tal efecto se utilizó un programa específico.

Modelo entidad-relación completo

El modelo entidad-relación de la Figura 23 nos permite representar las entidades relevantes de nuestro sistema de información así como sus relaciones y propiedades.

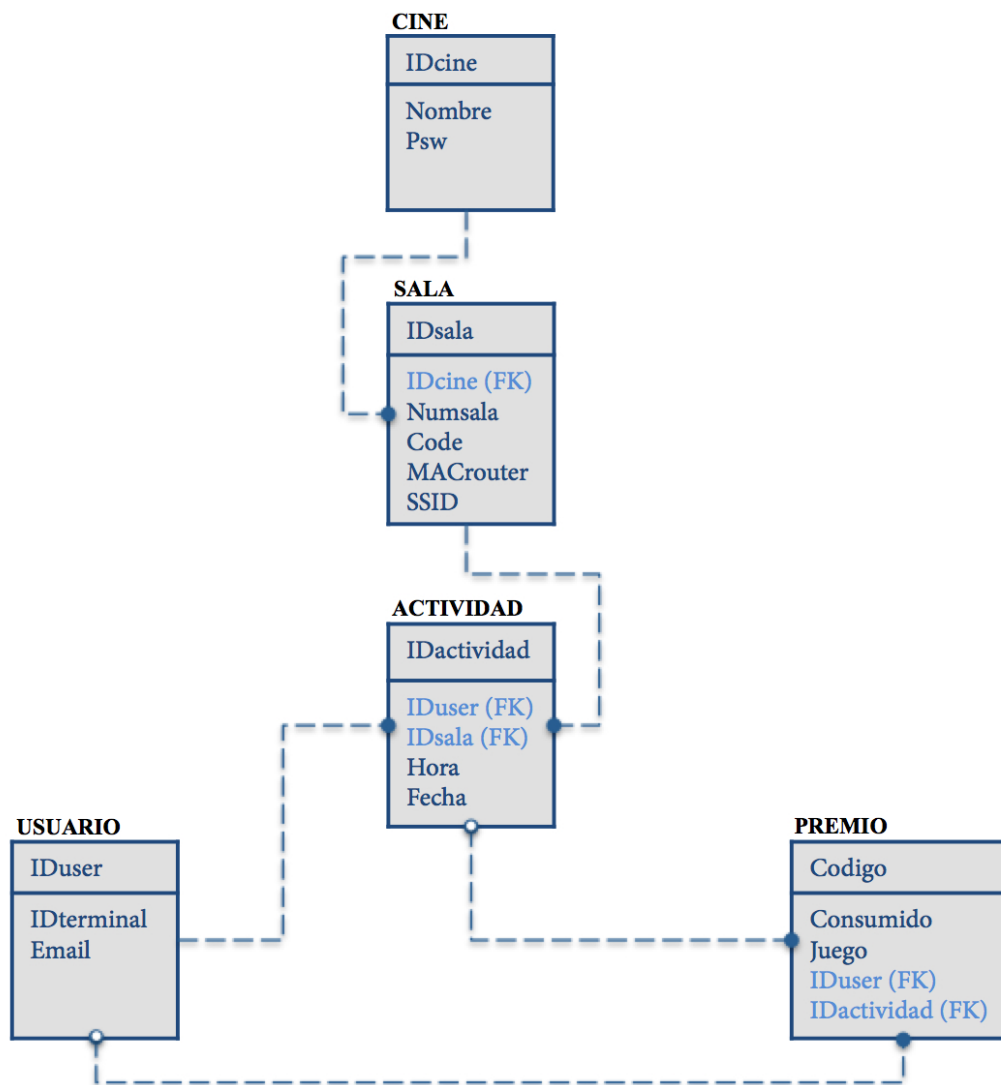


Figura 23. Modelo entidad relación base de datos.

Una vez realizado el diseño se pasó a su implementación, siendo necesario instalar MySQL. Para su instalación se descargó la versión gratuita del producto. Tras instalarlo y configurarlo correctamente se procedió a crear toda la estructura interna de la base de datos, por medio del lenguaje propio de MySQL. Se crearon todas las tablas con sus correspondientes atributos, y se especificaron las relaciones entre ellas.

Tablas y atributos

Como puede verse en la Figura 23, la base de datos se compone de 5 tablas, las cuales se detallan a continuación.

CINE

Atributo	Tipo	Otras características
IDcine (PK)	Integer	-Clave primaria -Autoincremento -No nulo
Nombre	Varchar(20)	-No nulo
Psw	Varchar(30)	-No nulo

Tabla 8. Atributos tabla cine.

SALA

Atributo	Tipo	Otras características
IDSala (PK)	Integer	-Clave primaria -Autoincremento -No nulo
IDcine (FK)	Integer	-Clave foránea -No nulo
Numsala	Integer	-No nulo
Code	Varchar(20)	-Nulo
MACrouter	Varchar(30)	-Nulo
SSID	Varchar(20)	-Nulo

Tabla 9. Atributos tabla sala.

USUARIO

Atributo	Tipo	Otras características
IDuser (PK)	Integer	-Clave primaria -Autoincremento -No nulo
IDterminal	Varchar(20)	-No nulo
Email	Varchar(30)	-Nulo

Tabla 10. Atributos tabla usuario.

ACTIVIDAD

Atributo	Tipo	Otras características
IDactividad (PK)	Integer	-Clave primaria -Autoincremento -No nulo
IDuser (FK)	Integer	-Clave foránea -No nulo
IDSala (FK)	Integer	-Clave foránea -No nulo
Hora	Time	-No nulo
Fecha	Date	-No nulo

Tabla 11. Atributos tabla actividad.

PREMIO

Atributo	Tipo	Otras características
Código (PK)	Varchar(20)	-Clave primaria -No nulo
IDactividad (FK)	Integer	-Clave foránea -No nulo
IDuser (FK)	Integer	-Clave foránea -No nulo
Consumido	Varchar(5)	-No nulo
Juego	Varchar(10)	-No nulo

Tabla 12. Atributos tabla premio.

Las tablas anteriores muestran en detalle las características y restricciones de los atributos que las componen.

Los elementos de cada tabla se identifican por su clave primaria (PK). En el caso de las tablas Cine, Sala, Usuario y Actividad, se trata de un campo de tipo entero que se asigna automáticamente al introducir un nuevo elemento. Por otro lado, en la tabla Premio, su clave primaria es el código asociado a cada premio. Todas estas claves cumplen la restricción de que tienen que ser únicas y no nulas.

En las tablas Sala, Actividad y Premio, hay campos marcados como claves foráneas (FK). Estos atributos se utilizan para referenciar a otras tablas. Son las claves de esas tablas y especifican así las relaciones entre ellas.

Relaciones

En la Figura 23 pueden apreciarse gráficamente las relaciones existentes entre las tablas de nuestra base de datos. Estas relaciones son las siguientes:

- Cine-Sala: Un cine esta compuesto de varias salas, de esta forma en la tabla correspondiente a la información de cada sala tenemos el identificador del cine al que pertenece.
- Sala-Actividad: Las actividades se desarrollan en salas de cine activas en el sistema Whatscine. De esta forma, entre la información de cada actividad, tenemos un campo que identifica la sala en la que se produjo.
- Usuario-Actividad: Cada actividad ha sido realizada por un usuario. Dentro de la tabla Actividad tenemos el identificador del usuario que la realizó.
- Premio-Usuario: Los premios se entregan al usuario que ha resultado ganador de un juego. Por este motivo, entre la información del premio se encuentra el identificador del usuario que lo ha recibido.
- Premio-Actividad: Los premios se otorgan tras el desarrollo de algún juego en una sala de cine. De esta forma, se asocia el premio a la actividad cuando se obtuvo, evitando almacenar información redundante dentro de la base datos. El identificador de la actividad queda almacenado dentro de la tabla premio.

Todas estas relaciones están configuradas para tener borrado en cascada. Esta decisión se tomó porque no resultaba útil para el sistema tener información descolgada.

Conexión

Al instalar MySQL queda como un elemento independiente en el ordenador. Por este motivo es necesario realizar una conexión entre la base de datos y el resto del servidor.

El servidor está escrito en lenguaje de programación Java. Este lenguaje no proporciona por si mismo los métodos necesarios para conectar el código con la base de datos MySQL. Para solucionar este problema manejamos dos posibles soluciones. La primera solución consistía en utilizar las opciones que nos proporcionaba el entorno de desarrollo NetBeans para configurar la conexión y manejar posteriormente la base de datos. La segunda opción se basaba en el uso de un Driver externo que había que añadir al proyecto.

Finalmente se optó por utilizar el driver proporcionado directamente por Oracle, para la conexión a bases de datos MySQL desde código Java.

mysql-connector-java-5.0.5-bin.jar

El primer paso tras obtener el driver consiste en importarlo en nuestro proyecto, agregándolo así al conjunto de librerías de las que hacemos uso.

Dentro de nuestro programa debemos incluir la librería `java.sql.*`, donde encontraremos todos los métodos necesarios para el manejo de la base de datos. Además, necesitamos la clase `org.gjt.mm.mysql.Driver` que viene incluida en el driver de MySQL.

Dado que casi todos los métodos relativos al uso de bases de datos pueden lanzar la excepción `SQLException`, englobaremos dentro de `try-catch` el código necesario.

Para realizar la conexión con la base de datos lo primero que debemos hacer es comprobar que el Driver se inicializa y se registra. Una vez hecho esto, utilizamos el método `getConnection()` de la clase `DriverManager` para establecer la conexión. Estos pasos se realizan dentro del método `conectar()` del núcleo del servidor. El código que realiza lo explicado anteriormente puede verse en la Figura 24.

```
public Connection conectar() {  
    conn = null;  
    try {  
        Class.forName("org.gjt.mm.mysql.Driver");  
        conn = DriverManager.getConnection(url, login, password);  
    } catch (SQLException ex) {  
        System.out.println("Hubo un problema al intentar conectarse con la base de datos " + url);  
    } catch (ClassNotFoundException ex) {  
        System.out.println(ex);  
    }  
    return conn;  
}
```

Figura 24. Código conexión base de datos.

Como se puede apreciar, el método `getConnection()` necesita los siguientes tres parámetros para establecer la conexión:

- url: contiene la dirección de la base de datos con el siguiente formato:

"jdbc:mysql://localhost/whatscine"

- jdbc:mysql: necesario porque estamos utilizando un driver jdbc para MySQL para realizar la conexión.

CAPÍTULO 4: Análisis del sistema

- localhost: el servidor de base de datos, en nuestro caso, está en el mismo ordenador en el que se correrá el código Java. Aquí puede ponerse una IP o un nombre de máquina que esté en la red.
 - whatscine: es el nombre de la base de datos que se ha creado dentro de MySQL.
- login: nombre de usuario para acceder a la base de datos.
 - password: contraseña asociada al usuario para acceder a la base de datos.

Con esto ya tendremos establecida la conexión. Para realizar cualquier acción (consultar, insertar nuevos registros, modificar los existentes o borrar), necesitamos una clase Statement proporcionada por la conexión. De esta clase utilizamos los métodos que nos permiten realizar las acciones en la base de datos. La forma de realizar estas acciones es mediante el paso de parámetros de tipo String, conteniendo los comandos MySQL. En la Figura 25 se puede ver un ejemplo de código para realizar una consulta a la base de datos.

```
stm = conn.createStatement();  
stm.executeQuery("SELECT IDsala FROM SALA WHERE IDcine = " + IDcine + " AND Numsala = " + numSala);
```

Figura 25. Ejemplo código consulta base de datos.

Los resultados a las consultas se obtienen a través de un ResultSet, el cual habría que leer en busca de la información requerida.

Una vez realizadas las operaciones necesarias, se cierra la conexión con la base de datos.

Núcleo

El núcleo del servidor es el bloque donde se procesa toda la información. Está escrito en el lenguaje de programación Java y se compone de dos clases: ConectarBD y Mail.

ConectarBD

Esta clase se encarga de procesar la información y gestionar la base de datos.

La información llega al servidor por dos vías diferentes, el servicio web y la página web. Como se ha explicado anteriormente, la única funcionalidad de estos bloques es la del intercambio de información, ya sea con los dispositivos móviles o con los usuarios de la página web. El resto de la funcionalidad que ofrecen se realiza mediante los métodos implementados en esta clase. En la Tabla 13 se muestra un resumen de estos métodos.

Nombre	ConectarDB	Tipo	Clase
Atributos	<ul style="list-style-type: none">String bd: Nombre de nuestra base de datos.String login: Nombre de usuario para acceder a nuestra base de datos.String password: Contraseña para acceder a nuestra base de datos.String url: Dirección donde esta alojada la base de datos.Statement stm: Objeto del tipo Statement que nos permite realizar operaciones en la base de datos.Connection conn: Objeto del tipo Connection que nos permite establecer la conexión con la base de datos.int IDCine: Número de identificación del cine.String cine: Nombre del cine.String nombreJuego: Lista con los juegos activos en el sistema.		
Métodos	consultarSala(String numSala)		
	Descripción	Método para obtener la información de una sala.	
	Tipo	String[]	
	Insertar3G(String código, String id, String hora, String fecha)		
	Descripción	Método para registrar una actividad cuando en la sala no hay red wifi.	
	Tipo	String	
	insertarWifi(String MACrouter, String SSID, String id, String hora, String fecha)		
	Descripción	Método para registrar una actividad cuando en la sala hay red wifi.	
	Tipo	boolean	
	Conectar()		
	Descripción	Método para establecer la conexión con la base de datos	
	Tipo	Connection	
	insertarSalas(String numSala , String code, String MACrouter, String SSID)		
	Descripción	Método para registrar una nueva sala de cine.	
	Tipo	String	
	modificarSala(String numSala , String code, String MACrouter, String SSID)		
	Descripción	Método para modificar la información de una sala de cine.	
	Tipo	String	
	getCine()		
	Descripción	Método para obtener el nombre del cine.	
Tipo	String		
nombreCine(String Cine)			
Descripción	Método para establecer el nombre del cine entre		

		los atributos de la clase.
	Tipo	Void
	login(String cine, String psw)	
	Descripción	Método para realizar el inicio de sesión en el sistema.
	Tipo	Boolean
	setCine(String cine, String psw)	
	Descripción	Método para registrar un nuevo cine en el sistema.
	Tipo	String
	getSalas()	
	Descripción	Método para obtener todas las salas de un cine, junto con toda información de cada una.
	Tipo	ArrayList<String[]>
	eliminarSala(String numSala)	
	Descripción	Método para borrar una sala de cine del sistema.
	Tipo	String
	ganador(String mail, String id, String hora, String fecha, String cGanador)	
	Descripción	Método para registrar a los usuarios ganadores y sus premios.
	Tipo	String
	mandarMail(String mail, String cGanador)	
	Descripción	Método que hace uso de la clase Mail para el envío de correos electrónicos a los ganadores.
	Tipo	void
	cifrar(String psw)	
	Descripción	Método que cifra la contraseña de inicio de sesión.
	Tipo	String
	gestionPremio(String código)	
	Descripción	Método encargado de gestionar la entrega de los premios.
	Tipo	String
	algoritmoPremio(String código)	
	Descripción	Método encargado de comprobar la veracidad de un código de premio.
	Tipo	boolean

Tabla 13. Clase ConectarBD.

En la mayoría de los métodos, el parámetro de salida es de tipo String. Esto se debe a que en muchos casos lo que se devuelve es un mensaje informando de cual ha sido el resultado del método.

Cuando se insertan o modifican elementos dentro de la base de datos, los métodos encargados realizan una serie de comprobaciones. Esto se hace para controlar que se cumplen las restricciones impuestas por el sistema. Estas restricciones, como se va a explicar a continuación, varían en función del tipo de elemento que se esté registrando.

- Cine:
 - No pueden existir dos cines con el mismo nombre. Además hay que tener en cuenta que el sistema no diferencia entre mayúsculas y minúsculas.
- Sala:
 - Cada sala tiene que contener el identificador del cine al que pertenece y su número dentro de ese cine.
 - No puede haber dos salas con el mismo número dentro del mismo cine.
 - No pueden existir dos sala con la misma dirección MAC asociada.
 - No puede haber dos salas con la misma SSID dentro del mismo cine.
 - No puede coincidir dos salas con el mismo código.
 - Cada sala tiene que contener los campos MAC y SSID rellenos o vacíos a la vez.
 - La sala tiene que contener al menos el código y/o la MAC y el SSID.
- Actividad:
 - Cada elemento actividad tiene que contener el identificador del usuario, la fecha, la hora y la sala donde se realizó.
 - Tienen que contener los campos código o MAC y SSID. Además estos han de pertenecer a salas ya registradas en el sistema.
- Usuario:
 - Cada elemento usuario tiene que contener el identificador de su terminal, y este no puede ser igual para dos usuarios distintos.
 - Dos usuarios no pueden tener la misma dirección de correo electrónico.
- Premio:
 - Cada elemento premio tiene que contener su código asociado, y además, este tiene que ser único y correcto.
 - Tiene que contener los identificadores del usuario y la actividad asociados.
 - Cada premio tiene que marcar si ha sido consumido o no, e indicar a que juego corresponde.

De entre los métodos que aparecen en la Tabla 13, se ha creído conveniente explicar uno de ellos con mayor detalle debido a que para su funcionamiento se hacen uso de elementos particulares. Como se describe en dicha tabla, el método cifrar() se encarga, como su propio nombre indica, de cifrar la contraseña utilizada para el inicio de sesión a través de la página web. Este método utiliza dos librerías proporcionadas por Java que tienen que ser importadas en el código, estas librerías son `java.security.spec.KeySpec` y `javax.crypto.*`. El cifrado que realiza el método se realiza por medio del algoritmo DES, que hace uso de una clave secreta de cifrado.

Tras realizar el cifrado de la contraseña, esta se codifica en Base64 de forma que pueda ser representada por caracteres imprimibles de ASCII.

Mail

Se trata de una clase auxiliar dentro del núcleo del servidor. La clase Mail se encarga del envío de correos electrónicos desde una cuenta del sistema Whatscine.

Los métodos implementados dentro de esta clase, para su correcto funcionamiento, pueden verse en la Tabla 14.

Nombre	Mail	Tipo	Clase
Atributos	<ul style="list-style-type: none">String user: cuenta de correo electrónico del sistema.String pass: contraseña para el acceso a la cuenta.String destino: dirección de correo electrónico del destino.String subject: Asunto del correo electrónico a enviar.String mensaje: Mensaje que se envía en el correo electrónico.Properties props: Objeto de tipo Properties para la configuración de la cuenta.		
Métodos	setGmailProps()		
	Descripción	Método para establecer la propiedades necesarias para el uso de la cuenta de correo electrónico.	
	Tipo	void	
	Send()		
	Descripción	Método encargado del envío de los correos electrónicos.	
	Tipo	boolean	

Tabla 14. Clase Mail.

Para el envío de correos electrónicos es necesario primero configurar la cuenta. Para este fin se utiliza el método setGmailProps(). En nuestro caso, la cuenta de correo electrónico del sistema Whatscine pertenece al servicio Gmail, por lo que las propiedades que tenemos que ajustar son las que aparecen en la Figura 26.

```
// El host de correo, en nuestro caso Gmail
props.setProperty("mail.smtp.host", "smtp.gmail.com");
props.setProperty("mail.smtp.starttls.enable", "true");
//El puerto que vamos a usar
props.setProperty("mail.smtp.port", "587");
//El usuario
props.setProperty("mail.smtp.user", user);
//Le indicamos que es necesario autenticarse
props.setProperty("mail.smtp.auth", "true");
```

Figura 26. Propiedades servicio Gmail.

Una vez especificadas las propiedades se elabora el mensaje, para lo cual hay que seguir los siguientes pasos:

1. Creamos un objeto de la clase Session, que representa nuestra conexión con el servidor de correo y le indicamos las propiedades de la Figura 26.

2. Se elabora el mensaje a enviar utilizando la sesión creada. Para ello creamos un objeto de la clase `MimeMessage` y le ponemos los datos que se indican a continuación:
 - a. Remitente
 - b. Destinatario
 - c. Asunto
 - d. Cuerpo del mensaje. En nuestro caso, al especificar el formato de texto que se va a enviar, especificamos que es del tipo “text/html”, el cual nos da mayores opciones que el texto plano.
3. Para enviar el mensaje utilizamos la clase `Transport`, especificando el protocolo a utilizar, en nuestro caso `smtp`.
4. Se establece la conexión utilizando el usuario y la contraseña de la cuenta del sistema.
5. Enviamos el mensaje.
6. Cerramos la conexión.

Los métodos y clases utilizados para el manejo de cuentas de correo electrónico vienen facilitados por la librería `javax.mail.*`, a excepción de la clase `Properties`, que se encuentra en `java.util.Properties`.

4.3 Servidor sala

Los servidores que se instalan en cada sala de cine ya estaban desarrollados antes del comienzo de este proyecto. Sin embargo, para conseguir la funcionalidad completa que se quería obtener era necesario que el servidor llevara a cabo nuevas tareas. De esta forma, durante el desarrollo del presente proyecto se tuvo que implementar un nuevo bloque de código que lleva a cabo estas funciones y que se integrará posteriormente en el sistema.

La tarea que era necesario que desarrollara el servidor era la de generar códigos para los usuarios que resultaran premiados (tarea 2). Para tal fin se desarrolló la función “código”, que ejecuta un algoritmo de codificación para generarlo.

El lenguaje de programación utilizado es C, esto se debe a que es en el que se encuentra programado el servidor.

A continuación se dará una visión general del funcionamiento del algoritmo, sin entrar demasiado en detalle por tratarse de una parte secreta del sistema.

A grandes rasgos el algoritmo funciona como se muestra en el diagrama de la Figura 27.

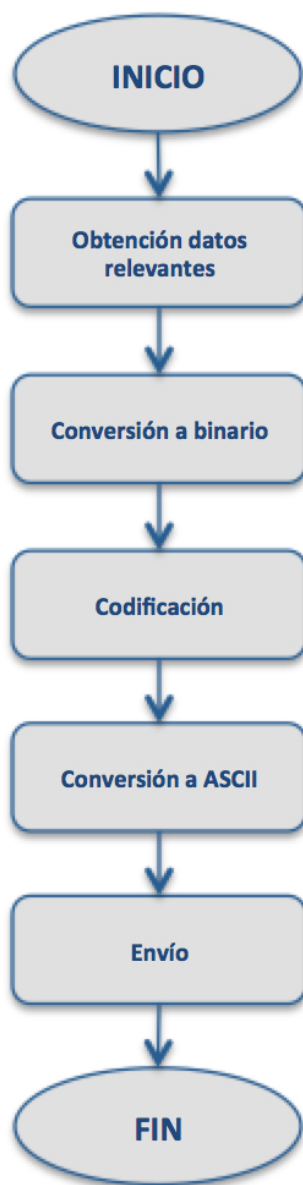


Figura 27. Diagrama del algoritmo de generación de códigos.

Ahora se explicarán los pasos mostrados en la Figura 27.

1. La función obtiene la información que es considerada relevante en la gestión del premio, y que será utilizada para generar el código. Los datos que se utilizan son:
 - La dirección MAC del router instalado en la sala donde se ha obtenido el premio.
 - El juego en el que el espectador resultó ganador. Los distintos juegos estarán identificados por un número dentro del sistema.
 - La fecha y hora de la sesión en la que se desarrolló el juego.

2. Una vez se tenga toda la información se hará una conversión de la misma a código binario.
3. Con todos los datos en código binario se procede a codificarlo mediante el algoritmo implementado. La codificación se realiza mediante la mezcla de los datos, colocando la información de tal forma que el receptor sea capaz de obtenerla. Se trata de esta forma de una codificación reversible.
4. Una vez realizada la codificación, convertimos el código binario a un código de 8 caracteres ASCII.
5. Tras la generación del código, se procede a su envío al dispositivo móvil. Esta parte no ha sido implementada durante el proyecto debido a que es necesario conocer el funcionamiento interno del servidor instalado en cada sala de cine. Será un paso que se desarrollará posteriormente, cuando se proceda a integrar la función dentro del sistema.

A continuación se muestra un ejemplo de código resultante de este algoritmo.

X9aEuP2A

El código se compone de un total de 8 caracteres ASCII. Durante la codificación se reservaron 8 bits para cada uno de estos caracteres, siendo la mitad de estos bits información útil para el sistema. De esta forma, en el código hay un total de 32 bits reservados para información del sistema.

Uno de los requisitos que tenía que cumplir el código es que fuera capaz de ser único para cada premio. Esto se consigue gracias a la información que se utiliza para generarlo, como la fecha, la sala, la hora y el juego. También era necesario comprobar el número de combinaciones posibles que se podía tener con un código de estas características. Tras realizar el cálculo teniendo en cuenta todas las restricciones impuestas por el código ASCII, la información utilizada y nuestro propio algoritmo, tenemos un total de más de 1400 millones de códigos posibles.

4.4 Cliente Android

Cada espectador que acude a una sala de cine, y quiere hacer uso de las funcionalidades de Whatscine, tiene que tener la aplicación instalada.

El módulo ha sido desarrollado para la versión Android de la aplicación, utilizando íntegramente el lenguaje de programación Java adaptado para los requisitos del sistema operativo.

La estructura de clases del modulo puede verse en la Figura 28.

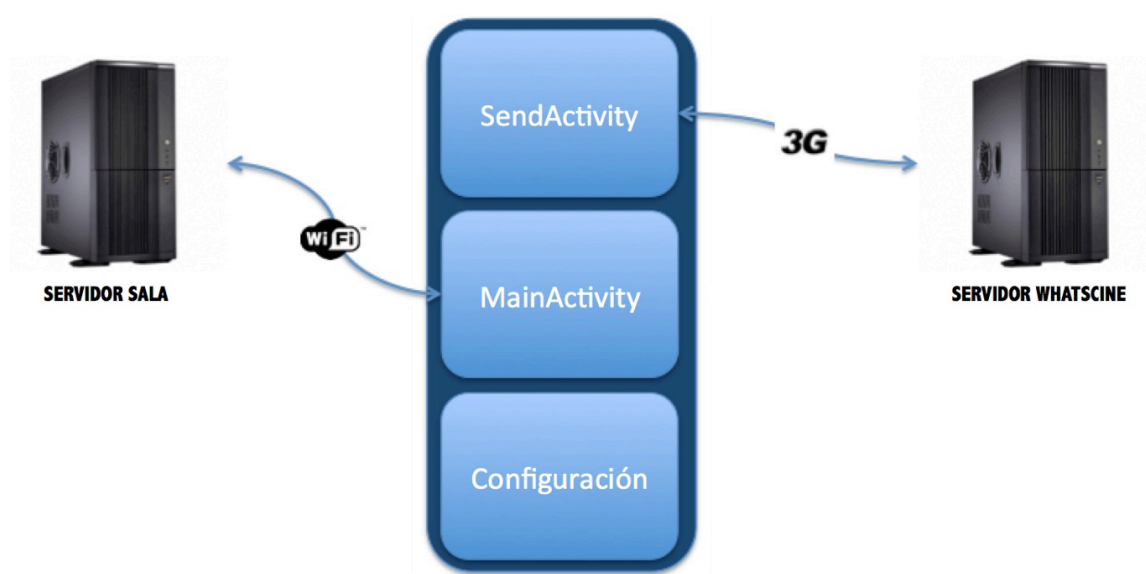


Figura 28. Estructura de clases del módulo Android.

A continuación se explicará en detalle cada una de las clases y demás componentes del cliente Android implementado en el proyecto.

Clases

MainActivity

Esta clase es el núcleo funcional del módulo, donde se procesa la información y se realizan la mayor parte de las funciones. Para algunas tareas concretas se apoya en las otras dos clases de las que se compone el sistema.

A continuación voy a explicar como se han implementado las diferentes operaciones que realiza esta clase.

- Comprobación del estado de red

La aplicación tiene que comprobar el estado de la conectividad de red del dispositivo móvil, de esta forma sabe si está conectado a la red wifi de la sala de cine o si dispone de acceso mediante redes móviles a internet. Estas comprobaciones las realiza gracias a la clase “ConnectivityManager”, que se encuentra disponible en el API de Android. Esta clase nos facilita la información del estado de las redes disponibles en el dispositivo, además de notificar cuando se produce algún cambio.

- Gestión datos servidores

Otra de las operaciones que realiza la clase principal es gestionar las información que tiene que intercambiar con los dos servidores de los que se compone el sistema.

La aplicación utiliza distintos métodos de comunicación para cada uno de los servidores. Cuando se trata del servidor propio de Whatscine, se hace uso de la clase auxiliar “SendActivity” cuyo funcionamiento se explica más adelante. En el caso de

los servidores de las salas de cine, el método del intercambio de información está sin definir ya que su implementación se realizara cuando se integre el sistema realizado en este proyecto dentro del sistema completo de Whatscine, ajustándose a los mecanismos que ya estén en funcionamiento en la aplicación. Por otro lado, los parámetros necesarios para identificar las salas de cine se obtienen directamente del router por medio de la clase “WifiManager”, que proporciona el API necesario para el manejo de todos los aspectos de la conexión wifi.

Entre la información que se tiene que enviar al servidor de Whatscine se encuentra la fecha y hora, en la que se realizan ciertas operaciones, y el número de serie del terminal, que servirá como identificador único de cada dispositivo. Para la obtención de estos datos se utilizan clases y métodos proporcionados por la API de Android. En el caso del número de serie se hace uso de la clase “TelephonyManager”.

- Gestión de códigos de premio

Como se ha comentado anteriormente, en el servidor de las salas de cine se genera un código para los ganadores de los distintos juegos. Este código se transmite desde dichos servidores hasta el dispositivo de cada ganador. La clase MainActivity se encarga de realizar las gestiones necesarias con dicho código. Las operaciones que se realizan con el código una vez el dispositivo resulte ganador de algún premio son:

- Envío del código al servidor de Whatscine, tras el registro o comprobación de que el usuario ya estaba registrado.
- Almacenamiento en el dispositivo. Para esta operación se utiliza la clase “configuración” que se explicara más adelante. El código queda almacenado en el terminal hasta que vuelva a resultar ganador de algún premio.
- Despliegue de una ventana informativa en la aplicación, avisando de que ha resultado ganador y mostrando el código del premio.

- Registro usuario

El registro del usuario se hace necesario una vez que el dispositivo a resultado ganador por primera vez. Para realizar dicho registro hay que introducir una dirección de correo electrónico válida, que se pedirá mediante una ventana emergente en la aplicación. Dicha dirección quedará almacenada de forma persistente, para lo cual se hará uso de la clase “configuración”.

Una vez el usuario haya registrado una dirección de correo electrónico, esta no podrá ser modificada ni eliminada, salvo que se desinstale la aplicación, de forma que si vuelve a instalarse, el sistema volverá a requerir un registro la primera vez que resulte ganador.

Para comprobar la validez de los correos electrónicos proporcionados por los usuarios ha sido necesario implementar un método que compare el correo con un patrón, evitando así que pueda escribirse cualquier combinación de caracteres que no se ajuste a un correo electrónico. El patrón con el que se compara es el siguiente:

```
Pattern.compile("^[\w\.-]+\.[\w\.-]+*@[A-Za-z0-9-]+\.[A-Za-z]{2,4}$");
```

Si el resultado no es correcto, el sistema volverá a mostrar una ventana, pidiendo que se introduzca el correo de nuevo.

SendActivity

Esta clase se encarga de realizar el intercambio de información con el servidor propio de Whatscine.

Para su correcta implementación es necesario que extienda la clase abstracta “AsyncTask”. Esta clase permite realizar operaciones en segundo plano y publicar resultados en el hilo principal del sistema sin ser necesaria la manipulación de hilos o controladores.

El uso de esta clase para gestionar las funciones de envío y recepción de datos entre el cliente y el servidor se hace necesario para liberar de carga al hilo principal de la aplicación. Estas operaciones, que pueden ser de larga duración, podrían provocar efectos de lentitud, mal funcionamiento o incluso bloqueo cuando se realizan en el hilo principal, problemas que se solucionan realizando las operaciones en segundo plano.

Como se ha explicado anteriormente, el servidor de Whatscine basa sus comunicaciones en un servicio web. Android no dispone de comunicación nativa con este tipo de servicios, siendo necesario realizarlo manualmente o mediante librerías externas. De esta forma la implementación de la funcionalidad en esta clase se basa en el uso de una librería externa llamada “ksoap2-android” en su versión 3.0. Esta librería no viene incluida dentro del kit de desarrollo de Android, de forma que es necesario descargar el fichero con extensión “.jar” e incluirlo dentro del proyecto que se está desarrollando en Eclipse. Su funcionamiento se basa en el protocolo SOAP. Este protocolo define como dos objetos en diferentes procesos, pueden comunicarse por medio de intercambio de datos XML.

Para el correcto funcionamiento de la comunicación por medio de la librería ksoap2, es necesario ajustar los parámetros referidos al servicio web del servidor, tal y como se muestra en la Tabla 15.

Parámetro	Valor
<i>NAMESPACE</i>	http://Cine.ws.uc3m/
<i>ENDPOINT</i>	http://163.117.135.82:8080/WSCine/WebServiceCine
<i>METHOD</i>	“Conexión3G”, “conexionWIFI”, “correo”
<i>SOAPACTION</i>	NAMESPACE + METHOD

Tabla 15. Parámetros de configuración librería ksoap2.

Una vez configurados dichos parámetros, se procede a realizar la petición al servicio web. Esta petición se compone de 5 pasos:

1. Definir la petición.
2. Configurar el contenido del mensaje y como procesarlo.
3. Definir el canal de transporte.
4. Realizar la llamada.
5. Recoger los datos.

Hay que destacar que el paso 2 se realiza en conjunción con la clase principal, ya que es donde se prepara el contenido del mensaje a enviar.

Una vez recogidos los datos estos son procesados en la clase principal.

Configuración

Clase encargada de almacenar de forma persistente datos necesarios para el funcionamiento del módulo. Para la implementación de esta clase se ha utilizado la interfaz “SharedPreferences” que provee el API de Android, cuya utilidad es guardar información de preferencia útil para la aplicación, dando la opción de acceder y modificar la misma.

Los datos almacenados en este caso son el código generado cuando el usuario del dispositivo resulta ganador de un premio y el email que el usuario utiliza para registrarse en el sistema de Whastcine.

Interfaz

Una vez que el módulo desarrollado durante este proyecto se integre en la aplicación de Whastcine para Android realizará todas sus funciones en un segundo plano, de manera transparente al usuario, con las excepciones de unas pocas ventanas emergentes con avisos. Por este motivo no ha sido necesario desarrollar una interfaz gráfica para el resultado final. Sin embargo, para la realización de pruebas de funcionamiento, se implementó una sencilla interfaz donde activar las diferentes funciones del sistema, pudiendo de esta forma simular situaciones parecidas a las que se podrían dar en la realidad.

Permisos

Para que una aplicación desarrollada para el sistema operativo Android funcione correctamente, es necesario dotarla de algunos permisos para realizar ciertas operaciones. Estos permisos hay que especificarlos en el archivo “AndroidManifest.xml” que se encuentra dentro del proyecto de desarrollo de la aplicación.

Los permisos necesarios para este proyecto pueden verse en la Tabla 16.

Permiso Android	Descripción
ACCESS_NETWORK_STATE	Permite el acceso a información sobre las redes.
ACCESS_WIFI_STATE	Permite el acceso a información sobre el estado de la red WIFI.
INTERNET	Permite la apertura de sockets de red.
READ_PHONE_STATE	Permite el acceso de solo lectura a la información de estado del teléfono.

Tabla 16. Permisos necesarios para el módulo Android.

SDK

Android es una plataforma de software libre, por lo que cuenta con un SDK disponible para todo desarrollador que lo desee, que incluye, entre otros elementos, el conjunto completo de API que este sistema soporta. En el caso concreto de nuestra

CAPÍTULO 4: Análisis del sistema

aplicación, la versión del API utilizada para el desarrollo ha sido la 16, siendo la 8 la versión mínima necesaria para su funcionamiento. En la Figura 29 se muestra el “Android SDK Manager”, con el cual se gestionan las versiones de API que disponemos.

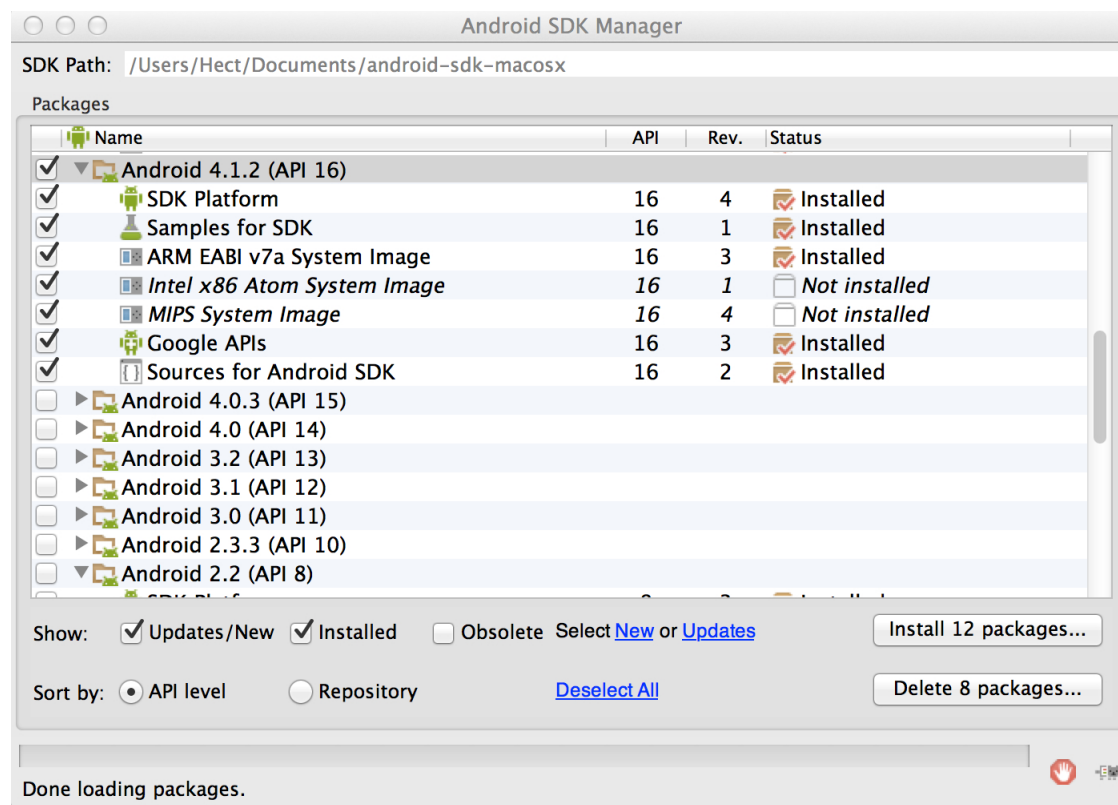


Figura 29. SDK Android.

El SDK puede descargarse directamente de la web de Android. Una vez descargado es necesario referenciarlo desde el entorno de desarrollo, en nuestro caso Eclipse. Para lograrlo hay que instalar un plug-in específico de Android para la plataforma Eclipse. Esta herramienta, llamada ADT, facilita la creación de proyectos, su implementación, depuración y ejecución.

Finalmente hay que indicar en las preferencias de Eclipse la localización del SDK previamente descargado.

Llegados a este punto, ya se tiene el entorno listo para desarrollar la aplicación Android.

4.5 Pruebas

Una vez finalizada la fase de desarrollo del proyecto se realizaron diversas pruebas del sistema completo.

Debido a que el sistema se compone en su mayor parte de bloques o módulos que serán integrados en sistemas mayores ya funcionales, nos encontramos con diversas limitaciones a la hora de realizar las pruebas.

Se explicarán detalladamente los procedimientos que se llevaron a cabo para probar los distintos componentes del sistema, junto con los resultados obtenidos. Se dividirán en apartados según tuvieron que probarse de forma independiente o de manera conjunta.

Para cada una de las pruebas de aceptación del sistema se elabora una tabla donde se muestra el informe de los resultados. Estas pruebas se realizan por y para los usuarios, y tienen como objetivo validar formalmente que el sistema se ajusta a sus necesidades.

Algoritmo de generación de códigos

En el caso del algoritmo de generación de códigos, el cual se encuentra totalmente aislado del resto de componentes, fue necesario probarlo de forma independiente. Se realizaron las pruebas introduciendo de forma manual la información que necesitaba, operación que posteriormente tendrá que hacer automáticamente en conjunción con el resto del servidor. Una vez introducidos los datos se ejecutaba el algoritmo, obteniendo códigos distintos y correctamente formados según cambiábamos los datos de entrada. No fue posible realizar el envío automático al dispositivo móvil, ya que como se comentó en el apartado correspondiente, esta parte no ha podido ser implementada.

Identificador	P-01
Descripción	Generación de códigos con distinta información de origen.
Resultado	Correcto

Tabla 17. P-01.

Página Web

La página web, alojada en el servidor Whatscine, podía probarse de manera independiente al resto del sistema. Para ello se lanzaba el servidor y se accedía por medio de un navegador web. Se realizaron pruebas de todas las funciones habilitadas.

Identificador	P-02
Descripción	Registro de un nuevo usuario.
Resultado	Correcto. Mensaje: "Registro realizado con éxito".

Tabla 18. P-02.

Identificador	P-03
Descripción	Registro con nombre de usuario en uso.
Resultado	Correcto. Mensaje: "Ya existe un cine con ese nombre".

Tabla 19. P-03.

Identificador	P-04
Descripción	Registro con algún campo en blanco.
Resultado	Correcto. Mensaje: "Introduce un nombre de cine valido", "Introduce una contraseña valida".

Tabla 20. P-04.

Identificador	P-05
Descripción	Registro con contraseña diferente en los dos campos.
Resultado	Correcto. Mensaje: "Las contraseñas no coinciden".

Tabla 21. P-05.

Identificador	P-06
Descripción	Inicio de sesión.
Resultado	Correcto.

Tabla 22. P-06.

Identificador	P-07
Descripción	Inicio de sesión con usuario y/o contraseña incorrectos.
Resultado	Correcto. Mensaje: "Usuario o contraseña incorrectos".

Tabla 23. P-07.

Identificador	P-08
Descripción	Insertar nueva sala.
Resultado	Correcto. Mensaje: "Insertado con éxito".

Tabla 24. P-08.

Identificador	P-09
Descripción	Insertar nueva sala con número en uso dentro del mismo cine.
Resultado	Correcto. Mensaje: "Ya existe una sala con ese número en este cine".

Tabla 25. P-09.

Identificador	P-10
Descripción	Insertar nueva sala con código en uso.
Resultado	Correcto. Mensaje: "Ya existe una sala con ese código".

Tabla 26. P-10.

Identificador	P-11
Descripción	Insertar nueva sala con MAC en uso
Resultado	Correcto. Mensaje: "Ya existe una sala con esa MAC".

Tabla 27. P-11.

Identificador	P-12
Descripción	Insertar nueva sala con SSID en uso dentro del mismo cine.
Resultado	Correcto. Mensaje: "Ya existe una sala en este cine con esa SSID".

Tabla 28. P-12.

Identificador	P-13
Descripción	Insertar nueva sala con campos MAC y SSID no rellenos o vacíos a la vez.
Resultado	Correcto. Mensaje: "La sala tiene que contener los campos MAC y SSID rellenos o vacíos a la vez".

Tabla 29. P-13.

Identificador	P-14
Descripción	Insertar nueva sala sin información suficiente.
Resultado	Correcto. Mensaje: "La sala tiene que contener la información de la red wifi o el código de sala".

Tabla 30. P-14.

Identificador	P-15
Descripción	Modificar información sala.
Resultado	Correcto. Mensaje: "Información sala modificada".

Tabla 31. P-15.

Identificador	P-16
Descripción	Modificar información sala con código en uso.
Resultado	Correcto. Mensaje: "Ya existe una sala con ese código".

Tabla 32. P-16.

Identificador	P-17
Descripción	Modificar información sala con MAC en uso.
Resultado	Correcto. Mensaje: "Ya existe una sala con esa MAC asociada".

Tabla 33. P-17.

Identificador	P-18
Descripción	Modificar información sala con SSID en uso en el mismo cine.
Resultado	Correcto. Mensaje: "Ya existe una sala en este cine con esa SSID".

Tabla 34. P-18.

Identificador	P-19
Descripción	Modificar información sala con campos MAC y SSID no rellenos o vacíos a la vez.
Resultado	Correcto. Mensaje: "La sala tiene que contener los campos MAC y SSID rellenos o vacíos a la vez".

Tabla 35. P-19.

Identificador	P-20
Descripción	Modificar información sala sin información suficiente.
Resultado	Correcto. Mensaje: "La sala tiene que contener la información de la red wifi o el código de sala".

Tabla 36. P-20.

Identificador	P-21
Descripción	Eliminar sala.
Resultado	Correcto.

	Mensaje: "Sala eliminada".
--	----------------------------

Tabla 37. P-21.

Identificador	P-22
Descripción	Comprobar código premio correcto.
Resultado	Correcto. Mensaje: "El código proporcionado es correcto, puede entregar el premio".

Tabla 38. P-22.

Identificador	P-23
Descripción	Comprobar código premio mal formado.
Resultado	Correcto. Mensaje: "El código proporcionado no es correcto".

Tabla 39. P-23.

Identificador	P-24
Descripción	Comprobar código bien formado pero sin premiar.
Resultado	Correcto. Mensaje: "El código proporcionado no corresponde a ningún premio".

Tabla 40. P-24.

Identificador	P-25
Descripción	Comprobar código de premio ya entregado.
Resultado	Correcto. Mensaje: "El código proporcionado ya ha sido utilizado".

Tabla 41. P-25.

Todas las situaciones expuestas anteriormente tuvieron resultados esperados y positivos, realizando las funciones cuando todo era correcto, y mostrando mensajes con los errores correspondientes cuando no lo era.

Aplicación Android + Servidor Whatscine

Para comprobar el correcto funcionamiento del módulo desarrollado para la aplicación Android de Whatscine, ha sido necesario crear una aplicación completa que lo integrara, cuya funcionalidad se basa en la del módulo. Esto se hacía necesario para poder instalarlo y ejecutarlo dentro de un dispositivo móvil real, ya que el módulo está diseñado para funcionar en un segundo plano dentro de la aplicación Whatscine, siendo de esta forma transparente para el usuario, que solo interactuará con él cuando se requiera algún dato que no pueda obtenerse de manera automática, y siempre por medio de ventanas emergentes. Para esta aplicación de prueba, se

CAPÍTULO 4: Análisis del sistema

diseñó una interfaz de usuario básica que incluía ciertos componentes que permitían el uso de la misma.

En la Figura 30 puede verse una captura de pantalla de la aplicación de prueba desarrollada.

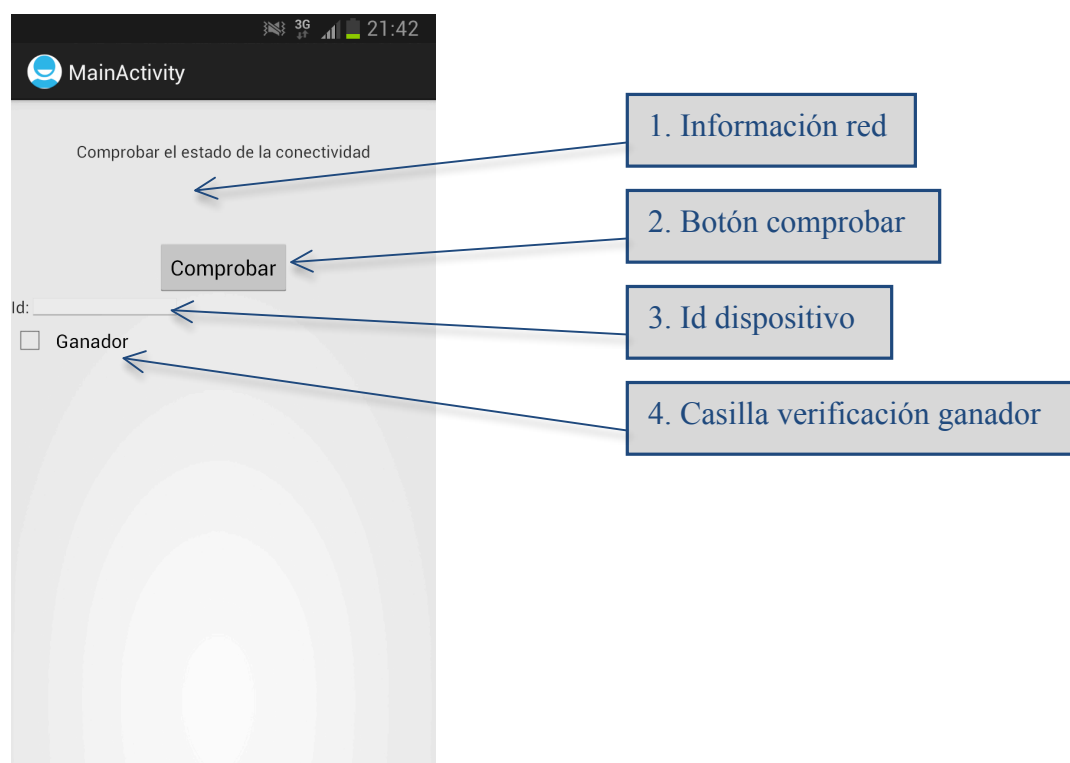


Figura 30. Vista aplicación de prueba.

La vista de la aplicación se compone de cuatro elementos.

1. Cuadro de texto donde se muestra la información referente al estado de la conectividad de red. Nos mostrará un mensaje indicando si el dispositivo se encuentra conectado por wifi, 3g o sin ninguna conexión. En caso de estar conectado por wifi, nos mostrará la MAC del router y la SSID de la red.
2. Botón que activa la comprobación del estado de la conectividad, y de esta forma el resto de la funcionalidad de la aplicación.
3. Cuadro de texto donde se muestra el número de identificación del dispositivo. Siendo este el número de serie del dispositivo
4. Casilla que nos permite indicar a la aplicación que el dispositivo ha resultado ganador. Esto nos permite probar la aplicación tanto si el usuario ha ganado, como si no lo ha hecho.

La funcionalidad de la aplicación Android no se puede probar por completo sin el servidor de Whatscine, es por esto que las pruebas se realizaron haciendo funcionar de manera conjunta la aplicación y el servidor Whatscine desarrollado.

Dentro del servidor se harán uso de tres de los cuatro bloques que lo conforman, el servicio web, el núcleo y la base de datos.

Para realizar las pruebas se montó el servidor en un ordenador portátil, conectado a internet por cable Ethernet y una dirección de ip única. Por otro lado se instaló la

aplicación en dos dispositivos con sistema operativo Android. Cada uno de los dispositivos tenía instalada una versión distinta de Android, por un lado un Samsung Galaxy S3 con Android 4.1.2 y por otro un Samsung Galaxy Ace con Android 2.2. De esta forma se abarca un gran rango de versiones del sistema operativo. Tras el montaje se realizaron varias pruebas con diferentes escenarios que podían darse en la realidad.

Identificador	P-26
Descripción	Sala de cine con red wifi y usuario no premiado.
Resultado	Correcto.

Tabla 42. P-26.

Identificador	P-27
Descripción	Sala de cine con red wifi y con usuario premiado y no registrado.
Resultado	Correcto.

Tabla 43. P-27.

Identificador	P-28
Descripción	Sala de cine con red wifi y con usuario premiado y registrado.
Resultado	Correcto.

Tabla 44. P-28.

Identificador	P-29
Descripción	Sala de cine sin red wifi y con usuario no premiado.
Resultado	Correcto.

Tabla 45. P-29.

Identificador	P-30
Descripción	Sala de cine sin red wifi y con usuario premiado y no registrado.
Resultado	Correcto.

Tabla 46. P-30.

Identificador	P-31
Descripción	Sala de cine sin red wifi y con usuario premiado y registrado.
Resultado	Correcto.

Tabla 47. P-31.

Identificador	P-32
Descripción	Introducir un correo electrónico con estructura incorrecta.
Resultado	Correcto. Mensaje: "Formato email incorrecto".

Tabla 48. P-32.

Identificador	P-33
Descripción	Introducir un correo electrónico ya registrado en el sistema.
Resultado	Correcto. Mensaje: "Email ya registrado, introduzca uno nuevo".

Tabla 49. P-33.

Los resultados obtenidos demostraban que el sistema funcionaba correctamente de forma conjunta. Tras realizar las pruebas expuestas anteriormente, se podía apreciar en la base de datos que se realizaba el registro de la actividad de forma correcta. Por otro lado, cuando se marcaba al usuario como ganador, se realizaba también un registro del premio, y se recibía un correo electrónico con el código en la dirección facilitada. En este caso, dado que no existe conexión con el algoritmo de códigos, el código se introducía en la aplicación de forma manual tras ejecutar el algoritmo de forma independiente.

En la Figura 31 puede verse un ejemplo de un correo electrónico recibido tras ganar un premio jugando en el sistema Whatscine.

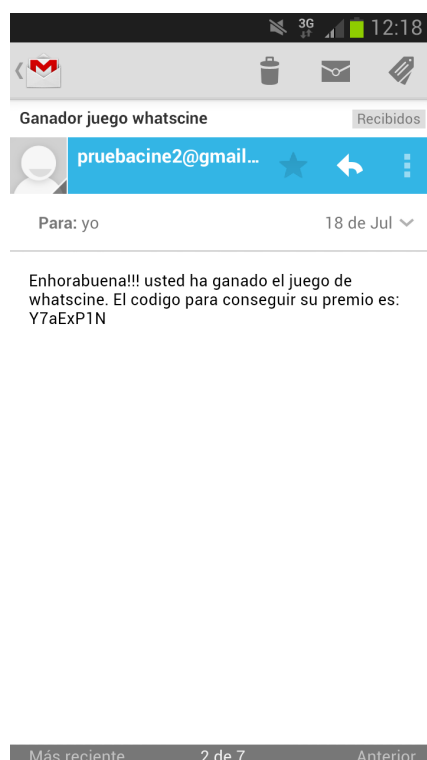


Figura 31. Ejemplo de correo recibido premiado

Capítulo 5

Presupuesto

En este apartado se realiza un resumen de las fases de desarrollo por las que ha pasado el proyecto y se muestra un desglose de los costes estimados que conlleva su implementación.

5.1 Fases de desarrollo

El primer paso que se llevó a cabo al comenzar el proyecto, fue establecer las especificaciones y requisitos que se tenían que cumplir. Una vez fijados los objetivos, se buscaron posibles soluciones y se planificó el desarrollo del proyecto en varias fases.

En primer lugar se diseñó lo que sería la arquitectura general que tendría el sistema a implementar, ajustándose las distintas fases de desarrollo a los componentes que lo forman.

Fase 1: Servidor Whastcine

El servidor Whastcine fue el primer componente que se desarrolló en el sistema. Esta decisión se tomó porque era donde residía la mayor parte de la funcionalidad, siendo de esta forma el bloque mas complejo del proyecto. Además el cliente Android necesita el servidor para funcionar, haciéndose imprescindible para realizar pruebas.

Había que tener en cuenta también que resultaba mas sencillo ajustar las comunicaciones en función del servidor.

El desarrollo del servidor también se dividió en función de los tres bloques que lo forman. En primer lugar se procedió a la implementación de la base de datos, que podía considerarse como la parte mas independiente del servidor, ya que no requería el uso del resto de bloques para funcionar. Una vez finalizada se comenzó con el desarrollo del núcleo, el cual marcaba la funcionalidad de los bloques restantes, y hacia uso de la base de datos ya implementada. El siguiente paso fue la implementación de la página web, la cual funcionaba en conjunción con el núcleo. La última parte que se desarrolló del servidor fue el servicio web, dado que resultaba mucho mas sencillo plantear las comunicaciones necesarias en función de la información requerida.

El último paso de esta fase fue la realización de pruebas del servidor, solventándose los errores que pudieron aparecer. Dadas las características de los componentes que lo forman, los bloques sobre los que se realizaron las pruebas fueron la página web, el núcleo y la base de datos. Esto se debe a que podía probarse su funcionamiento de forma conjunta, dejando las pruebas del servicio web para cuando el cliente Android estuviera finalizado.

La duración total de esta fase fue finalmente de 31 días.

Fase 2: Cliente Android

La segunda fase dentro del desarrollo del proyecto fue la implementación del cliente Android. Esta fase se llevó a cabo en un total de 18 días y consistió en el diseño del módulo con las funcionalidades necesarias para la aplicación de Whatscine, su adaptación a una aplicación provisional con la que poder llevar a cabo las pruebas necesarias y finalmente su implementación.

Una vez finalizado el cliente Android, se realizaron pruebas de forma independiente para comprobar que cumplía con las especificaciones, y llevaba a cabo las funcionalidades necesarias. Tras solventar los errores surgidos en las pruebas, se procedió a configurar y establecer la conexión entre el cliente Android y el servidor de Whatscine. Esta parte resultó problemática debido a que las diferentes versiones del API de Android imponían distintas restricciones a la hora de establecer la conexión, forzándonos a encontrar un método funcional en todas las versiones.

Finalmente se llevaron a cabo las pruebas necesarias del conjunto cliente – servidor.

Fase 3: Servidor Sala

La tercera fase consistió en el desarrollo del algoritmo de generación de códigos, que posteriormente tendrá que integrarse en los servidores instalados en las salas de cine. Esta fase se realizó en tres pasos: el diseño del algoritmo, su implementación y finalmente las pruebas.

La duración total fue de 8 días, de los cuales 5 fueron dedicados a la parte de diseño. El diseño del algoritmo requirió de más tiempo de trabajo debido a que se tuvo que crear una codificación que cumpliera con unas características y restricciones muy estrictas.

Fase 4: Sistema Completo

La última fase de desarrollo consistió en la realización de pruebas del conjunto general del proyecto. Estas pruebas consistieron en exponer el sistema a posibles situaciones reales, tal y como se explicó en el apartado 4.5, titulado Pruebas, de este proyecto.

Durante esta fase, también se realizaron algunas mejoras y optimizaciones en el código.

Finalmente y tras realizar con éxito todas las fases de desarrollo, se comenzó a redactar esta memoria.

En la Figura 32 se expone el diagrama de Gantt del proyecto.

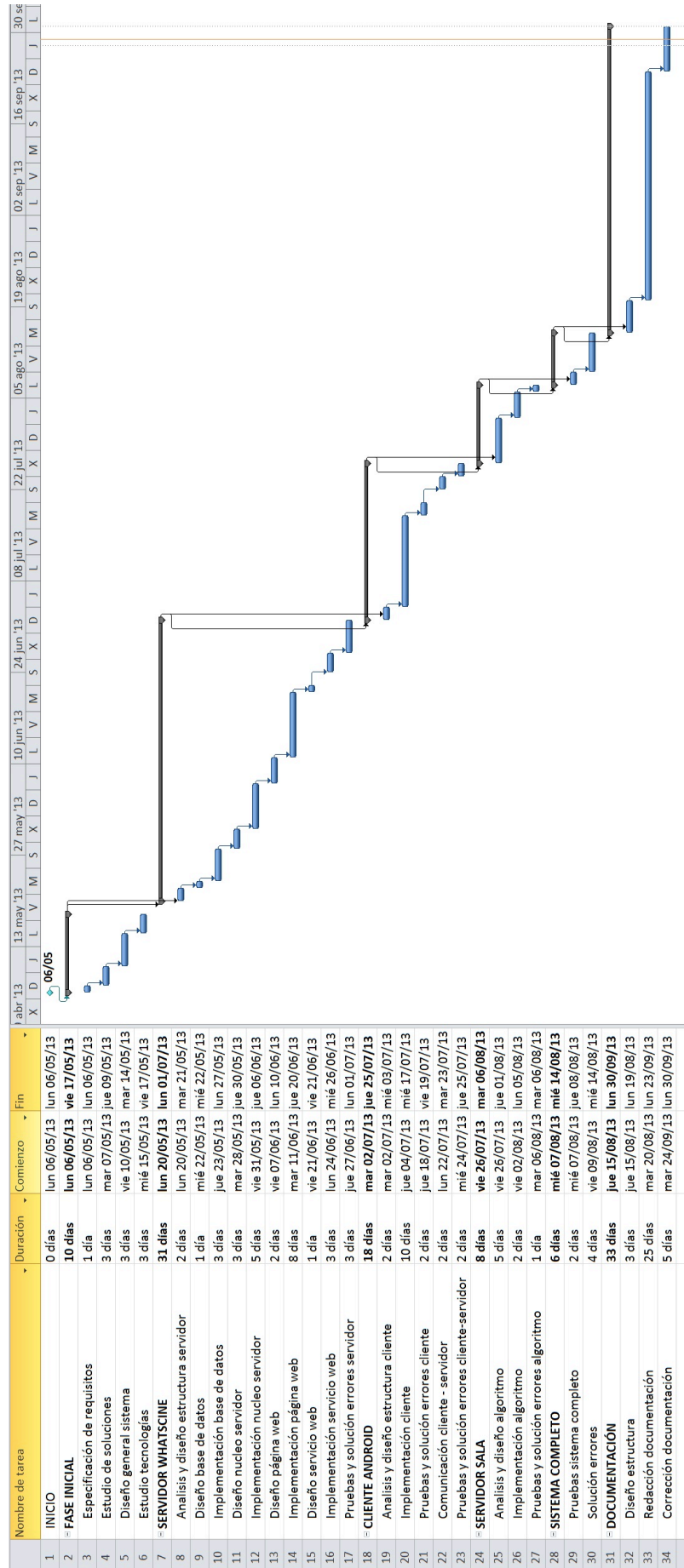


Figura 32. Diagrama de Gantt.

5.2 Desglose de costes

Para realizar el presupuesto se tienen en cuenta tres grupos de costes distintos. Por un lado tenemos los costes del personal involucrado en el desarrollo, por otro consideramos el material utilizado para las pruebas e implementación y finalmente se consideran los gastos directos derivados del proyecto.

En la realización del presupuesto se han tenido en cuenta los siguientes aspectos:

- Si no se especifica lo contrario, los costes que aparecen en el presupuesto se encuentran expresados en Euros.
- Las cantidades económicas que figuran a lo largo del siguiente documento han sido calculadas aplicando redondeo clásico a dos decimales.

El presupuesto detallado se muestra en la Figura 33.



UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:

Hector Molla Ruiz

2.- Departamento:

Informática

3.- Descripción del Proyecto:

- Título **Desarrollo de un sistema para el registro y control de actividades en salas de cine.**
- Duración (meses) **5**
- Tasa de costes indirectos: **20%**

4.- Presupuesto total del Proyecto (valores en Euros):

22.483,05 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación (hombres mes) ^{a)}	Coste hombre mes	Coste (Euro)
García Crespo, Ángel		Ingeniero Senior	1	4.289,54	4.289,54
Molla Ruiz, Hector		Ingeniero	5	2.694,39	13.471,95
Hombres mes 6				Total	17.761,49

^{a)} 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
MacBook Pro	1.700,00	100	5	60	141,67
Samsung Galaxy S3	499,00	100	5	60	41,58
Microsoft Project Standart 2013	769,00	100	1	60	12,82
Samsung Galaxy Ace	199,00	100	1	60	3,32
Total					199,38

^{d)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

OTROS COSTES DIRECTOS DEL PROYECTO^{e)}

Descripción	Empresa	Costes imputable
Internet	Movistar	450,00
Material de oficina		75,00
Electricidad		250,00
Total		775,00

^{e)} Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	17.761
Amortización	199
Costes de funcionamiento	775
Costes Indirectos	3.747
Total	22.483

Figura 33. Presupuesto del proyecto.

Capítulo 6

Conclusiones y líneas futuras

En este capítulo se detallarán cuales han sido las conclusiones obtenidas al finalizar el presente proyecto, así como las posibles líneas futuras y mejoras por las que se podría continuar.

6.1 Conclusiones

El rápido desarrollo que están sufriendo las tecnologías móviles, junto con su gran acogida y expansión, traen consigo un enorme abanico de posibilidades. Las ventajas proporcionadas por los dispositivos móviles, provocan que su uso sea cada vez mayor. El uso de estas tecnologías se integra en la mayoría de actividades diarias de la sociedad.

El sistema implementado en este proyecto se integra a la perfección dentro de una de las actividades sociales mas comunes, ir al cine. Su funcionalidad está pensada para ajustarse a la aplicación Whatscine. Mejora su experiencia de uso gracias a que proporciona un gran número de posibles líneas de expansión, de las cuales hablaremos en detalle en el siguiente apartado. La expansión y utilidad del sistema desarrollado va, de esta forma, directamente ligado al del sistema completo de Whatscine. Actualmente casi la totalidad de los usuarios acude a las salas de cine con algún dispositivo móvil, lo que junto a las virtudes del propio sistema, facilita su uso e integración.

El sistema desarrollado, una vez integrado en la aplicación final, trabajará en segundo plano, sin que el usuario tenga que realizar ninguna operación mas allá que actualizar la aplicación o facilitar cierta información de manera puntual. Esto supone una gran ventaja, ya que de cara al publico, no acarrea ninguna dificultad de uso, proporcionándole sin embargo nuevas experiencias como usuario.

Por otro lado, tenemos la experiencia de uso para las salas de cine. La página web se ha desarrollado para que su uso resulte lo mas sencillo e intuitivo posible. Finalmente se ha logrado que el manejo de la información por parte de las salas sea rápido, proporcionando todos los datos y opciones de una forma rápida y clara. Todas las herramientas necesarias para realizar las operaciones se encuentran al alcance de menos de dos simples pasos, siendo su uso perfectamente accesible para cualquiera.

El funcionamiento automático de la mayor parte del sistema, provoca que su mantenimiento sea de una gran sencillez.

En el aspecto personal, los 5 meses dedicados al proyecto han supuesto un gran enriquecimiento personal y profesional. El desarrollo se ha llevado a cabo en un inmejorable ambiente de trabajo, resultando así mas sencillo e instructivo. Uno de los puntos fuertes de este proyecto reside en la gran cantidad de diferentes tecnologías, herramientas y lenguajes de programación que han sido utilizados. Académicamente, este escenario resulta inmejorable. Me ha proporcionado una gran experiencia en el desarrollo de software, ampliando mis conocimientos en campos que me resultan de gran interés, como Android, pero no solo han sido conocimientos técnicos lo aprendido en estos 5 meses. Sino también a enfrentarme ante una situación real de un proyecto de gran envergadura, donde ha sido necesaria una planificación inicial para alcanzar unos objetivos en un tiempo determinado, aprendiendo a desenvolverme y ajustarme a las dificultades que se presentaron en el camino.

6.2 Líneas futuras

Una de las mayores utilidades proporcionada por este sistema es el gran número de opciones de ampliación que conlleva. A continuación se van a exponer las líneas posibles para continuar con el desarrollo del sistema, entre las que se incluyen las mejoras directas que podrían realizarse y las posibles ampliaciones que las funcionalidades del sistema proporciona.

- Sistema iOS. La primera y más evidente línea de futuro del sistema del proyecto es realizar su implementación para el sistema operativo iOS. El objetivo sería trasladar las funcionalidades implementadas en Android a los dispositivos móviles de Apple.
- Proceso de activación del registro. Cuando se gana un premio y se solicita una dirección de correo electrónico para realizar el registro, el sistema comprueba que el dato facilitado realmente se trata de una dirección bien formada, pero esta podría ser falsa o inventada. Una mejora sería añadir un proceso de activación, donde se compruebe que la dirección es real. Este proceso podría consistir en el envío mediante un correo a la dirección proporcionada de un código o de un link. De esta forma, la cuenta no sería funcional hasta su activación, quedando el envío del código para el premio pendiente.

GLOSARIO

- Geolocalización. Entre la información disponible de las salas podría añadirse sus coordenadas. De esta forma podría utilizarse la geolocalización como forma auxiliar de identificar donde se lleva a cabo la actividad. Además añadiría información muy relevante para otras posibles mejoras que se exponen a continuación.
- Puntuación películas. Disponiendo de la información relativa a la actividad, como son la fecha, la hora y la sala de cine, se puede saber cual ha sido la película que ha visto el usuario. De esta forma se puede desarrollar un código para que el espectador puntúe la película, guardando un registro de sus gustos.
- Recomendación películas. Este añadido sería una extensión de lo expuesto en el punto anterior. Gracias a la información obtenida sobre los gustos de cada usuario, podría implementarse un sistema de recomendación de películas. Además, con la opción de la geolocalización, podrían realizarse recomendaciones más completas, ya que no solo se propondría una película, sino que también una sesión concreta, gracias a que conocemos los horarios y cines que el usuario suele frecuentar.

Glosario

ADT	Android Development Tools
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
CSS	Cascading Style Sheets
DES	Data Encryption Standard
DCP	Digital Cinema Package
FK	Foreign Key
FTP	File Transfer Protocol
GCC GNU	Compiler Collection
GPL	General Public License
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JSP	JavaServer Pages
MAC	Media Access Control
OASIS	Organization for the Advancement of Structured Information Standards
PK	Primary Key
SDK	Software Development Kit
SMTP	Simple Mail Transfer Protocol
SOAP	Simple Object Access Protocol
SSID	Service Set Identifier
TMS	Theatre Management System
UDDI	Universal Description, Discovery and Integration
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
WS-I	Web Services Interoperability Organization
WS-Security	Web Service Security
XML	Extensible Markup Language
XML-RPC	XML Remote Procedure Call
SQL	Structured Query Language

Referencias

- [1] 08/09/2013). Definición y explicación del término servidor. Available: <http://www.masadelante.com/faqs/servidor>
- [2] 08/09/2013). ¿Que es un servidor? Available: <http://www.anerdata.com/que-es-un-servidor.html>
- [3] (2013, 10/08/2013). Dispositivo móvil. Available: http://es.wikipedia.org/wiki/Dispositivo_m%C3%B3vil
- [4] (2013, 10/08/2013). Teléfono inteligente. Available: http://es.wikipedia.org/wiki/Tel%C3%A9fono_inteligente
- [5] (2013, 10/08/2013). Tableta. Available: [http://es.wikipedia.org/wiki/Tableta_\(computadora\)](http://es.wikipedia.org/wiki/Tableta_(computadora))
- [6] K. WorldPanel. (2013, 05/09/2013). El 28% de los móviles en España son Nokia.
- [7] (2013, 10/08/2013). Aplicación móvil. Available: http://es.wikipedia.org/wiki/Aplicaci%C3%B3n_m%C3%B3vil
- [8] U. C. Madrid. (2013, 14/09/2013). <http://www.whatscine.es>.
- [9] Condesa. (2011, 24/07/2013). ¿Que es Android? Available: <http://androideity.com/2011/07/03/que-es-android/>
- [10] (10/09/2013). *Android Developers*. Available: <http://developer.android.com>
- [11] Á. J. Vico. (2011, 21/07/2013). Arquitectura de Android. Available: <http://columna80.wordpress.com/2011/02/17/arquitectura-de-android/>
- [12] Angel. (2011, 12/08/2013). Ciclo de vida de una aplicación Android. Available: <http://droideando.blogspot.com.es/2011/02/ciclo-de-vida-de-una-aplicacion-android.html>
- [13] Condesa. (2011, 20/07/2013). Procesos en Android. Available: <http://androideity.com/2011/07/06/procesos-en-android/>
- [14] Oracle. (15/07/2013). *Java*. Available: <http://www.java.com/>
- [15] TIOBE. (2013, 16/09/2013). Programming Community Index for September 2013. Available: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [16] (2013, 20/07/2013). Web Service. Available: http://es.wikipedia.org/wiki/Servicio_web

GLOSARIO

- [17] (23/07/2013). *JavaServer Pages Technology*. Available: <http://www.oracle.com/technetwork/java/overview-138580.html>
- [18] M. A. Alvarez. (2001, 24/07/2013). *Que es Html*. Available: <http://www.desarrolloweb.com/articulos/que-es-html.html>
- [19] 24/07/2013). *Html*. Available: <http://es.wikipedia.org/wiki/HTML>
- [20] 22/07/2013). *Guía breve de CSS*. Available: <http://www.w3c.es/Divulgacion/GuiasBreves/HojasEstilo>
- [21] . *MySQL Reference Manual*. Available: <http://dev.mysql.com/doc/refman/5.1/en/>
- [22] (2013, 11/09/2013). *Apache Tomcat*. Available: <http://tomcat.apache.org>
- [23] M. Banahan, D. Brady, and M. Doran, *The C book*, Second ed.: Addison Wesley, 1991.
- [24] (12/09/2013). *NetBeans IDE*. Available: <https://netbeans.org/features/index.html>
- [25] (2003, 20/07/2013). *Eclipse Platform Technical Overview*. Available: <http://eclipse.org/whitepapers/eclipse-overview.pdf>
- [26] 21/07/2013). *Xcode*. Available: <http://es.wikipedia.org/wiki/Xcode>